

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Alan Kravanja

Forenzični sistem izboljšave slik varnostnih kamer

DIPLOMSKO DELO

VISOKOŠOLSKI STROKOVNI ŠTUDIJSKI PROGRAM PRVE
STOPNJE RAČUNALNIŠTVO IN INFORMATIKA

Ljubljana, 2017

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Alan Kravanja

Forenzični sistem izboljšave slik varnostnih kamer

DIPLOMSKO DELO

VISOKOŠOLSKI STROKOVNI ŠTUDIJSKI PROGRAM PRVE
STOPNJE RAČUNALNIŠTVO IN INFORMATIKA

MENTOR: viš. pred. dr. Borut Batagelj

Ljubljana, 2017

To delo je ponujeno pod licenco *Creative Commons Priznanje avtorstva-Deljenje pod enakimi pogoji 2.5 Slovenija* (ali novejšo različico). To pomeni, da se tako besedilo, slike, grafi in druge sestavine dela kot tudi rezultati diplomskega dela lahko prosto distribuirajo, reproducirajo, uporabljajo, priobčujejo javnosti in predelujejo, pod pogojem, da se jasno in vidno navede avtorja in naslov tega dela in da se v primeru spremembe, preoblikovanja ali uporabe tega dela v svojem delu, lahko distribuira predelava le pod licenco, ki je enaka tej. Podrobnosti licence so dostopne na spletni strani creativecommons.si ali na Inštitutu za intelektualno lastnino, Streliška 1, 1000 Ljubljana.



Izvorna koda diplomskega dela, njeni rezultati in v ta namen razvita programska oprema je ponujena pod licenco *GNU General Public License*, različica 3 (ali novejša). To pomeni, da se lahko prosto distribuira in/ali predeluje pod njenimi pogoji. Podrobnosti licence so dostopne na spletni strani <http://www.gnu.org/licenses>.

Kazalo

Povzetek

Abstract

| | | |
|-------------------|--|-----------|
| Poglavje 1 | Uvod | 1 |
| Poglavje 2 | Težave pri izboljšavi slik..... | 3 |
| 2.1 | Digitalna slika | 3 |
| 2.2 | Primer: registrska tablica | 4 |
| 2.2.1 | Slaba ločljivost | 4 |
| 2.2.2 | Izgubna kompresija..... | 5 |
| 2.2.3 | Optična zameglitev | 5 |
| 2.2.4 | Zameglitev zaradi prekomernega premikanja objekta..... | 6 |
| 2.2.5 | Slaba osvetlitev in kontrast..... | 6 |
| 2.2.6 | Šum..... | 7 |
| 2.3 | Pregled obstoječe programske opreme | 8 |
| 2.3.1 | vReveal | 8 |
| 2.3.2 | Ikena Forensic | 9 |
| 2.3.3 | Amped FIVE | 9 |
| 2.3.4 | GIMP | 10 |
| 2.3.5 | Adobe Photoshop..... | 11 |
| Poglavje 3 | Tehnike izboljšav | 13 |
| 3.1 | Iskanje robov..... | 14 |
| 3.1.1 | Cannyjev algoritem | 15 |
| 3.2 | Odstranjevanje šuma..... | 17 |
| 3.2.1 | Gaussov šum..... | 17 |
| 3.3 | Ostrenje slike | 19 |
| 3.3.1 | Neostra maska | 19 |

| | | |
|-------------------|--|-----------|
| 3.4 | Spreminjanje svetlosti in kontrasta..... | 22 |
| 3.5 | Segmentacija slik..... | 23 |
| 3.6 | Superresolucija | 23 |
| 3.6.1 | Večokvirne superresolucijske metode | 24 |
| 3.6.1.1. | Interpolacijski pristop | 25 |
| 3.6.1.2 | Frekvenčno-domenski pristop | 26 |
| 3.6.1.3 | Regularizacijski pristop | 27 |
| 3.6.2 | Enoslikovne superresolucijske metode | 29 |
| Poglavje 4 | Implementacija superresolucije in rezultati | 31 |
| 4.1 | Rezultati bikubičnega interpolacijskega algoritma | 33 |
| 4.2 | Rezultati bilinearnega interpolacijskega algoritma | 37 |
| Poglavje 5 | Zaključek | 41 |
| Literatura | 43 | |

Seznam uporabljenih kratic

| kratica | angleško | slovensko |
|--------------|--|--|
| CFT | continuous Fourier transform | neprekinjena Fourierjeva transformacija |
| CLS | constrained least squares | omejeni najmanjši kvadrati |
| CMYK | cyan, magenta, yellow, key (black) | modra, vijolična, rumena, črna |
| CSI | crime scene investigation | preiskava prizorišča zločina |
| DFT | discrete Fourier transform | diskretna Fourierjeva transformacija |
| GIMP | GNU image manipulation program | program GNU za manipulacijo slik |
| PDF | probability density function | funkcija gostote verjetnosti |
| RAISR | rapid and accurate image super-resolution | hitra in natančna slikovna superresolucija |
| RGB | red, green, blue | rdeča, zelena, modra |
| SEEDS | superpixels extracted via energy-driven sampling | super piksli (slikovne točke), izvlečeni z energičnim vzorčenjem |

Povzetek

Naslov: Forenzični sistem izboljšave slik varnostnih kamer

V diplomskem delu je obravnavan problem izboljšave slik videonadzornih kamer v namene forenzičnih raziskav. V forenziki se tehniki pogosto srečujejo s problemom premajhne ločljivosti oziroma prevelike oddaljenosti nekega ključnega objekta na videoposnetku. V delu so predstavljeni obstoječa programska oprema za izboljšavo slik ter osnovni filtri in tehnike izboljšav, ki so lahko v pomoč. Diplomsko delo se nato osredotoči na bolj zapleteno metodo superresolucije. Problem se je skušal rešiti z algoritmi superresolucije, bolj natančno z bikubičnim in bilinearnim interpolacijskim pristopom. Proučevano je bilo, koliko superresolucija lahko pomaga v različnih primerih.

Ključne besede: superresolucija, algoritem, bikubični interpolacijski pristop, bilinearni interpolacijski pristop

Abstract

Title: Forensic system for surveillance camera image enhancement

In this thesis the problem of enhancing images of video surveillance cameras for purposes of forensic investigation will be addressed. In forensics technicians often come across the problem of low resolution, or the key subject being too far away on the video or image. First the software that allows technicians to process images and the basic filters and processing techniques will be presented. After that the thesis will focus on a more complex method called super-resolution. The problem of low resolution will try to be solved with super-resolution algorithms, more specifically using bicubic interpolation approach and bilinear interpolation approach. The main point of this thesis, is to learn how much the super-resolution can help in different scenarios.

Keywords: super-resolution, algorithm, bicubic interpolation approach, bilinear interpolation approach

Poglavje 1 Uvod

Pri izboljševanju slik in videoposnetkov v namene forenzičnih raziskav gre večinoma za obnovo in izboljšavo posnetkov varnostnih kamer, da bi s slike lahko razbrali čim več informacij. Pri izboljšanju slik velja pravilo – razberemo in izboljšamo lahko le tiste informacije, ki se na sliki dejansko nahajajo. »Filmske« tehnike izboljšav, ko s slik zelo slabe kakovosti naredimo sliko visoke ločljivosti, žal (še) ne obstajajo, čeprav prinašajo nekateri novejši pristopi zelo obetavne rezultate [1].

Največkrat pri izboljšanju ne delamo na celi sliki ali videu, temveč nas zanimajo le manjše ključne podrobnosti, kot so majhni gibi, oddaljena sprememba luči na semaforju, registrske tablice, obraz osumljenca in podobni detajli.

Forenzično procesiranje slik je pomembno za široko paleto poklicev, tako za policijo kot za detektive, zasebne preiskovalce, vojsko itd. V splošnem se forenzično procesiranje oziroma izboljšava slik in videoposnetkov uporablja za analizo nadzornih posnetkov s prizorišč zločinov. Lahko gre za rop banke, vlom, umor ali teroristični napad. Nadzorne kamere pa uporabljajo včasih tudi podjetja, da zagotovijo, da je vedenje delavcev v skladu s pravili. Z videonadzornimi sistemi se torej srečujemo povsod, in čeprav se analiza nekaterih nadzornih podatkov lahko opravi s programsko opremo, bo vedno potreba po človeškem analitiku.

Superresolucija je še najboljši približek filmskim tehnikam izboljšav, saj lahko s to metodo dosegamo precej dobre rezultate, seveda ob pravih pogojih. Metode superresolucije lahko grobo razdelimo na večslikovne metode – metode, ki združujejo več nizkoločljivostnih slik, pri čemer velja, da je vsaka slika pridobljena z manjšim zamikom slikovnih točk (to je pristop, na katerega se bomo osredotočili v tem diplomskem delu), in na metode, ki temeljijo na primerih (angl. *example based*) – učenje korespondence med nizko- in viskoločljivostnimi slikami [2].

V okviru diplomskega dela smo razvili grafični vmesnik, ki sprejme več nizkoločljivostnih slik, med katerimi je vsaka slika malenkost drugačna od drugih (obrnjena, zamaknjena). Na podlagi teh slik vmesnik omogoča izdelavo ene viskoločljivostne slike z uporabo superresolucije, in sicer z bilinearnim in bikubičnim interpolacijskim pristopom. Interpolacija deluje v treh korakih – to so registracija, interpolacija in izostritev.

V naslednjih poglavjih so najprej predstavljene težave, s katerimi se srečujejo forenziki pri izboljšavi slik. Za tem sledi predstavitev najpogostejših vzrokov za slabo sliko ali videoposnetek, ki nam preprečujejo, da bi razbrali ključne informacije in predstavitev nekaterih metod za izboljšavo. Na koncu predstavimo praktično uporabo in implementacijo interpolacijskega pristopa superresolucije v grafičnem vmesniku.

Poglavje 2 Težave pri izboljšavi slik

Pogosto se zaradi različnih razlogov dogaja, da je kakovost posnetkov nadzornih kamer zelo slaba. To lahko povzročijo slaba osvetlitev, slaba kakovost medija, šum, ki ga povzroča snemalna naprava, ali pogosto tudi prekomerno premikanje subjekta. Z različnimi filtri, odstranjevanjem šuma in s tehnikami izboljšav se v veliko primerih lahko s slike izvleče precej informacij.

Forenzična obdelava slik in videoposnetkov je izraz za metode in postopke izboljševanja digitalnih slik. Za doseganje tega cilja se največkrat uporablja digitalne filtre, ki lahko zmanjšajo šum v sliki, poskušajo izvleči čim več informacij iz senc na sliki, izostrijo sliko ... Razporeditev slikovnih pikslov – histogramov – se lahko prav tako uporablja za pridobivanje informacij s slik [4].

2.1 Digitalna slika

Sliko lahko opišemo kot dvodimenzionalno funkcijo $f(x,y)$. x in y sta v tem primeru koordinati slikovnih točk (angl. *pixels*), vrednost f na lokaciji (x,y) je intenziteta ali nivo sivine na tej točki. Intenziteta je direktno povezana s svetlostjo slikanega subjekta. Tehnike za forenzično obdelavo slik lahko torej interpretiramo kot transformacijo, ki jo izvedemo na vhodnem parametru (sliki), da dobimo končno sliko [3].

To lahko opišemo kot $g(x,y) = T[f(x,y)]$, kjer so:

- f – vhodna/začetna slika,
- x, y – koordinati pikslov,
- T – transformacijska metoda oziroma tehnika izboljšanja slike,
- g – končna izboljšana slika.

Poleg tehnik za forenzično obdelavo slik poznamo tudi tehnike za urejanje slik, kot npr. izrezovanje, spreminjanje velikosti, svetlost in kontrast, ki pa ne spadajo v kategorijo tehnik za forenzično obdelavo. Tehnike za forenzično obdelavo slik po navadi vključujejo spreminjanje vrednosti pikslov na tak ali drugačen način, medtem ko se to pri tehnikah za urejanje slik navadno ne zgodi [3].

2.2 Primer: registrska tablica

Eden najpogostejših primerov v računalniški forenziki je prepoznavanje registrske tablice avtomobila z nadzornega videoposnetka. Najpogostejši razlogi, da tablice ne vidimo, so slaba ločljivost posnetka, izgubno stiskanje – kompresija, slaba osvetlitev, zamegljena slika in seveda šum. Lahko se zgodi, da imamo vse te vzroke združene skupaj, včasih pa že en sam zadostuje, da s tablice ne moremo razbrati registrske oznake. Na sliki 2.1 vidimo primer dobro berljive registrske tablice [5].



Slika 2.1: Primer dobro berljive registrske tablice.

2.2.1 Slaba ločljivost

Ločljivost nam pove, koliko slikovnih točk imamo na sliki. Če nimamo dovolj točk, da bi z njimi narisali številko, potem te številke ne bomo videli. Slabo ločljivost prepoznamo tako, da približamo registrsko tablico in preštujemo, kako visoka je v pikslih. Če je višina na primer dve slikovni točki, je povsem nemogoče narisati številko s tema dvema slikovnima točkama. Primer nam prikazuje slika 2.2.

V tem primeru skorajda ni rešitve. V določenih situacijah pa se da težavo rešiti s preprosto digitalno povečavo (angl. *digital zoom*). Ta sicer ne doda nobenih informacij sliki, ampak le poveča na način, da so oblike in podrobnosti bolj prepoznavne očesu. Če imamo več okvirjev (posnetkov), lahko uporabimo tehnike za povprečevanje okvirjev (angl. *frame averaging*) ali superresolucijo.



Slika 2.2: Levo: slika slabe ločljivosti, desno: slika, izboljšana z digitalno povečavo.

2.2.2 Izgubna kompresija

Izgubna kompresija je glavna slabost digitalnih nadzornih sistemov, če jih primerjamo z videokasetnim rekorderjem. Tudi če je ločljivost dobra, lahko naredi kompresija neberljive številke (slika 2.3). Posnetki so stisnjeni zato, da imajo manjšo velikost, kar je pomembno pri shranjevanju ali pošiljanju posnetka. Pogost primer izgubne kompresije je, ko imamo sliko registrske tablice, pri kateri se zdi, da ko jo bomo povečali, bomo lahko videli številke, vendar se to ne zgodi.

Težava se lahko pojavlja na dva načina:

- lahko imamo dovolj pikslov, vendar je videti, kot da jih ima več enako vrednost, torej imamo realno veliko manjšo uporabno ločljivost, kot je dejanska ločljivost slike,
- pojavijo se podrobnosti, ki ne sodijo k sliki, ampak se pojavijo zaradi kompresije.

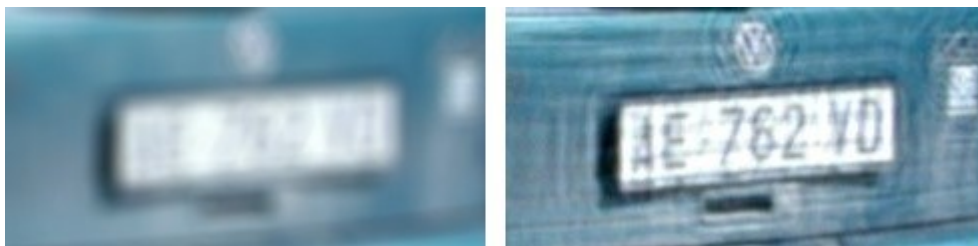
Pomagamo si lahko z določenimi filtri, kot je na primer deblokirni filter (angl. *deblocking filter*). Če imamo več okvirjev, lahko težavo učinkovito rešujemo s tehniko povprečevanja okvirjev (angl. *frame averaging*) ali superresolucijo (angl. *super resolution*).



Slika 2.3: Levo: rezultat izgubne kompresije, desno: slika, izboljšana z deblokirnim filtrom.

2.2.3 Optična zameglitev

Ta težava se pojavlja, ko slika ni povsem izostrena (slika 2.4). Po navadi so videonadzorne kamere konfigurirane tako, da imajo zelo veliko vidno polje. Vse je več ali manj v fokusu – izostreno, kar pa je zelo težko doseči. Če imamo dovolj ločljivosti, lahko težavo rešimo z optičnimi izostritvenimi filtri.

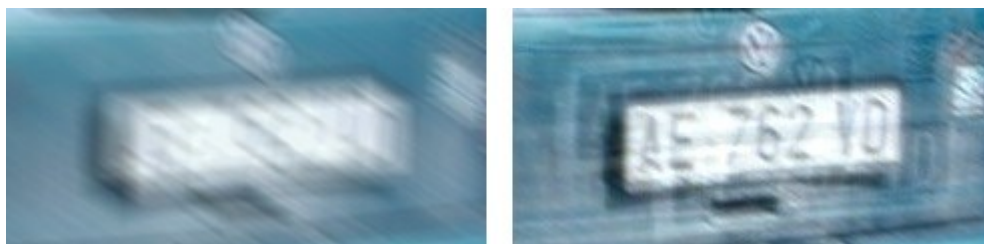


Slika 2.4: Levo: optična zameglitev, desno: slika, izboljšana z izostritvenimi filtri.

2.2.4 Zameglitev zaradi prekomernega premikanja objekta

Ta težava se pojavlja pri objektih, ki se prehitro premikajo v primerjavi s hitrostjo sprožilca kamere (slika 2.5). Učinek je zelo podoben optični zameglitvi, le da se tu točke spremenijo v črte namesto v kroge. Prepoznamo ga po tipični sledi, ki sledi točkam na sliki.

Če je to gibanje linearno, po navadi vzamemo neko referenčno točko, za katero vemo, da bi morala biti točka (npr. rob številke na registrski tablici), čeprav se na sliki pojavlja kot segment. Približno ocenimo razsežnosti zameglitve glede na dolžino in kot segmenta.

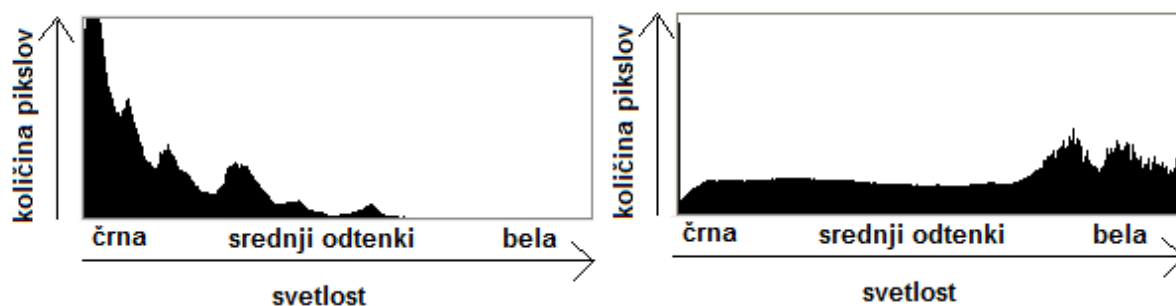


Slika 2.5: Levo: zameglitev zaradi premikanja objekta, desno: izboljšana slika.

2.2.5 Slaba osvetlitev in kontrast

Različni pogoji osvetlitve ali karakteristike kamere lahko povzročijo nastanek pretemnih (npr. slikanje ponoči) ali presvetlih (slikanje v smeri vira svetlobe) slik ali pa jim preprosto manjka kontrasta (vrednosti točk so si preveč podobne), kot prikazuje slika 2.7.

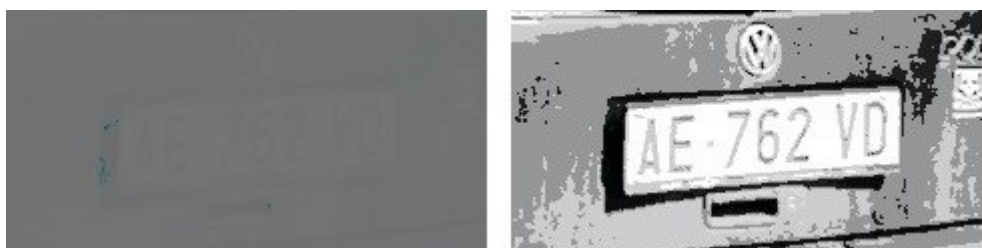
Težavo običajno enostavno prepoznamo, lahko pa si pomagamo s histogramom. Če je ta močno premaknjen proti levi, je slika pretemna, saj ima večina točk vrednosti okoli 0. Če je premaknjen proti desni, je slika presvetla – večina točk ima vednosti okoli 255. Prikaz histogramov vidimo na sliki 2.6.



Slika 2.6: Prikaz histogramov – levo je pretemna, desno pa presvetla slika.

Območje slike je prenasičeno, ko imajo vsi piksli vrednost 0 ali 255. Če je slika prenasičena, ne moremo storiti nič, ker tam ni več uporabnih informacij.

Težavo lahko rešimo na več načinov; najpogosteje ročno naravnamo svetlost in kontrast ali pa izravnamo histogram.

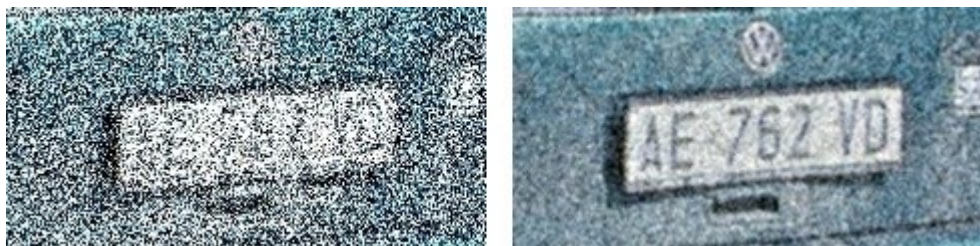


Slika 2.7: Levo: slika s slabim kontrastom, desno: slika po izravnavi histograma.

2.2.6 Šum

Šum je naključno odstopanje pikslov na sliki in je videti kot zrna, ki se običajno pojavijo na slikah, zajetih v slabi svetlobi (slika 2.8). Šum postane dobro viden, še posebej ko osvetlimo temen videoposnetek. Če imamo samo eno sliko, lahko izboljšamo kakovost z glajenjem (angl. *smoothing filter*). Včasih se pri tem zgodi, da filter odstrani tudi druge podrobnosti skupaj s šumom. Če imamo več okvirjev iste registrske tablice v istem položaju, si lahko pomagamo s filtrom povprečevanja okvirjev.

Šum lahko nastane na več različnih načinov, odvisno od tega, kako je slika nastala. Če je slika skenirana s fotografije, narejene na filmu, so lahko izvor šuma filmska zrna, poškodba filma ali pa šum nastane zaradi skenerja. V primeru, da je slika zajeta neposredno v digitalnem formatu, lahko nastane šum zaradi mehanizma za zajem slik. Šum pa lahko povzroči tudi elektronski prenos slike.



Slika 2.8: Levo: slika, popačena s šumom, desno: slika po glajenju.

2.3 Pregled obstoječe programske opreme

2.3.1 vReveal

vReveal je zelo osnoven program za izboljšavo posnetkov, katerega prvoten namen ni bila uporaba v forenzične namene. Uporablja zelo osnovne filtre za izboljšavo, kot so stabilizacija slike, izostritev, dodajanje kontrasta, poživitev barv ... Kljub temu pa lahko s programom dosežemo zelo dobre rezultate. S programom vReveal lahko uporabniki močno izboljšajo kakovost videoposnetkov, zajetih z mobilnimi telefoni in digitalnimi kamerami. Uporablja orodja, ki delujejo na en klik (angl. *one click tools*), za stabilizacijo, posvetlitev in izostritev popačenih posnetkov. Zelo dobro lahko izboljša podrobnosti v nizkoločljivostnih posnetkih in odstrani šum. Slika 2.9 prikazuje primer videa, ki je bil izboljšan s programom vReveal [7].



Slika 2.9: Izboljšava s programom vReveal – prej in potem.

Razlika je očitna. Vidimo, da sliko lahko zelo dobro izboljšamo že s preprostimi filtri, kljub temu pa s slike še vedno ni bilo mogoče razbrati ključnega dejanja, saj je bilo preveč oddaljeno.

2.3.2 Ikena Forensic

Ikena Forensic [8] je zelo napreden program, ki je v testni verziji precej omejen. Za izboljšavo uporablja bolj napredne algoritme, tudi superresolucijo. Primer izboljšave istega videa je prikazan na sliki 2.10.



Slika 2.10: Izboljšava s programom Ikena Forensic – prej in potem.

S programom Ikena Forensic lahko:

- izboljšujemo oziroma povečujemo obraze, ki so velikokrat ključna informacija, ki zanima forenzike: mnogokrat se podrobnosti izgubijo zaradi različnih dejavnikov, kot so sence, tresenje kamere, slaba osvetlitev ipd.;
- izboljšujemo oziroma povečujemo registrske tablice: te so pogosto tarča forenzikov. Uspešno branje tablice lahko pomeni razliko med ujetjem osumljenca in njegovim begom. Program Ikena Forensic to težavo reši s svetlobnim in kontrastnim filtrom ter lastnim patentiranim algoritmom superresolucije;
- obnovimo izgubljene podrobnosti: da izboljšamo sliko (okvir) z videoposnetka, patentiran superresolucijski algoritem analizira številne sosednje okvirje oziroma slike in izbere najboljše piksele. Iz njih nato sestavi novo sliko, ki je na koncu višje ločljivosti kot original. To nam omogoča, da dejansko izboljšamo ločljivost slike in dodamo nove podrobnosti, ki prej niso bile vidne, medtem ko znatno zmanjšamo nepravilnosti, ki so nastale zaradi kompresije.

2.3.3 Amped FIVE

Amped FIVE [9] je eden boljših programov, ki ne omogoča le izboljšave posnetkov in slik, ampak tudi merjenje dimenzij kraja zločina.

Program Amped FIVE omogoča:

- popravljanje perspektive objektov,
- merjenje višine iz videonadzornih posnetkov: program Amped FIVE lahko glede na ostale predmete na sliki zelo natančno določi višino osumljenca,
- ločevanje prstnega odtisa od ozadja,
- fokusiranje subjekta.

Program Amped FIVE je precej prilagodljiv glede izbire tehnologij za izboljšavo slik. Lahko izbiramo med osnovnimi tehnikami in filtri ali pa uporabljamo najnovejše vrhunske algoritme. Program avtomatsko generira poročila, tako da je preiskava jasno dokumentirana. Poročilo vsebuje vse tehnične podrobnosti in metode obdelave slik ter seveda ustrezne slike. Pri zelo dolgem videu nam program ponudi možnost samodejnega iskanja dogodkov. To v praksi pomeni, da če skušamo ugotoviti, kdaj je npr. osumljenec vstopil v banko, nam ni treba pogledati celotnega videa, ki lahko traja 12 ur ali več, ampak nam program sam poišče, kdaj se je ta dogodek zgodil glede na večji premik, ki se zgodi v posnetku, kot na primer vstop osumljenca v banko. Omogoča nam nalaganje slik iz večine bolj uporabljenih formatov, kot so jpg, png, bmp, tiff ... Prav tako podpira tudi večino videoformatov NTSC in izvora PAL ali videoposnetke z mobilnih telefonov in interneta. Ampedova notranja knjižnica omogoča predvajanje in dekodiranje videoposnetkov, tudi če nimamo nameščenih ustreznih kodekov. S programom lahko preverimo, če so bile na videu narejene kakšne nedovoljene spremembe, tako da pogledamo kodo datoteke EXIF in hash. To je pomembno za zagotavljanje pristnosti dokazov na sodišču.

2.3.4 GIMP

GIMP [10], [11] je prosto dostopen program za urejanje rasterskih slik, v osnovi namenjen za operacijski sistem GNU. Sicer ni namenjen izboljšavi slik v namene forenzičnih raziskav, ga pa velja omeniti, saj nam ravno tako omogoča nekatere funkcije, ki nam lahko pridejo prav, če nimamo dostopa do forenzičnega programa. S tem programom lahko retuširamo in urejamo slike, prosto rišemo, povečujemo, izvajamo rezanje, fotomontaže in podobna opravila. Program vključuje več kot 150 standardnih funkcij, med drugim tudi odstranjevanje zameglitve in šuma ter ostrenje slik. Pri tem uporablja tehniko ostrenja z neostro masko, kar nam lahko pride zelo prav, ko gre za forenzično izboljšavo slik. Program nam prav tako ponuja tako imenovana pametna orodja, ki uporabljajo bolj zapletene algoritme, ki bi bili drugače časovno zamudni ali celo nemogoči. Ta orodja vključujejo:

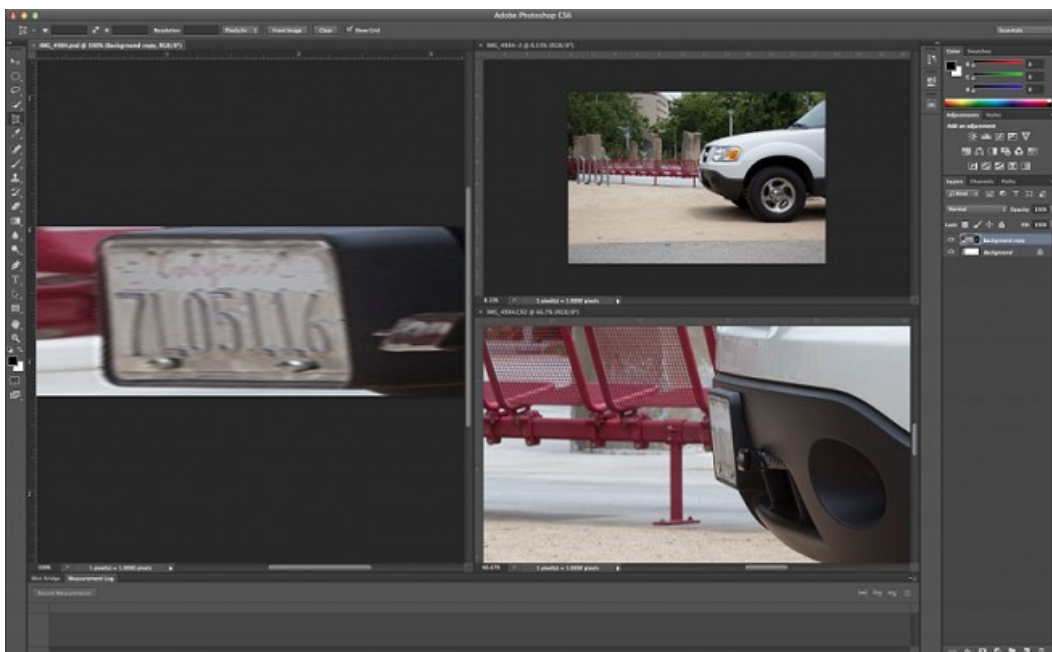
- orodje za kloniranje – klonira slikovno točko z uporabo čopiča,
- orodji za zameglitev in izostritev,

- zdravilni čopič – kopira slikovne točke z območja in popravlja ton in barvo,
- orodje, ki nam naredi ciljne slikovne točke temnejše ali svetlejše,
- korekcijski način omogoča popraviljanje perspektivnega popačenja, ki nastane zaradi nagiba objektiva.

2.3.5 Adobe Photoshop

Adobe Photoshop je najbolj znan in razširjen program za obdelavo rasterskih slik. Tako kot program GIMP tudi Photoshop ni bil namenjen za forenzične namene, vendar se zaradi svoje zmogljivosti pogosto uporablja na tem področju. Ustvarjen je bil leta 1988 in do danes postal industrijski standard v urejanju rasterske grafike. Z njim lahko urejamo slike na več plasteh, podpira alfa sestavljanje (angl. *alpha compositioning*), maske in večbarvnih modelov, kot so RGB, CMYK, dvotonski model (angl. *duotone*) in barvni model Lab (angl. *Lab color space*).

Forenzični tehniki ga pogosto uporabljajo za obdelavo velikih količin fotografskih dokazov, ki izvirajo s fotografij prizorišč zločinov. Kot večina drugih forenzičnih programov nam omogoča obdelavo prstnih odtisov, ostrenje slik, zmanjševanje šuma in popraviljanje perspektive slik (primer na sliki 2.11). Ima možnosti superresolucije v tem smislu, da omogoča povečavo slik z bilinearno in bikubično interpolacijo, vključuje pa seveda tudi klasične filtre za prilagajanje svetlosti, kontrasta ipd. [12].



Slika 2.11: Popraviljanje perspektive.

Poglavje 3 Tehnike izboljšav

Izboljšava, kot jo vidimo na sliki 3.1, v resnici ni mogoča, saj noben filter ali algoritem ne more dodati toliko manjkajočih informacij sliki. Lahko pa z različnimi filtri in algoritmi sliko izboljšamo in z nje razberemo čim več informacij, ki so bile morda skrite oziroma poškodovane s strani različnih faktorjev, kot so šum, kompresija in podobno.

Nekaj najpogostejših tehnik izboljšav:

- odstranitev šuma,
- stabilizacija slike,
- izostritev,
- dodajanje kontrasta,
- poživitev barv,
- segmentacija slik (angl. *image segmentation*),
- superresolucija,
- detekcija robov.



Slika 3.1: Primer filmske izboljšave iz serije CSI.

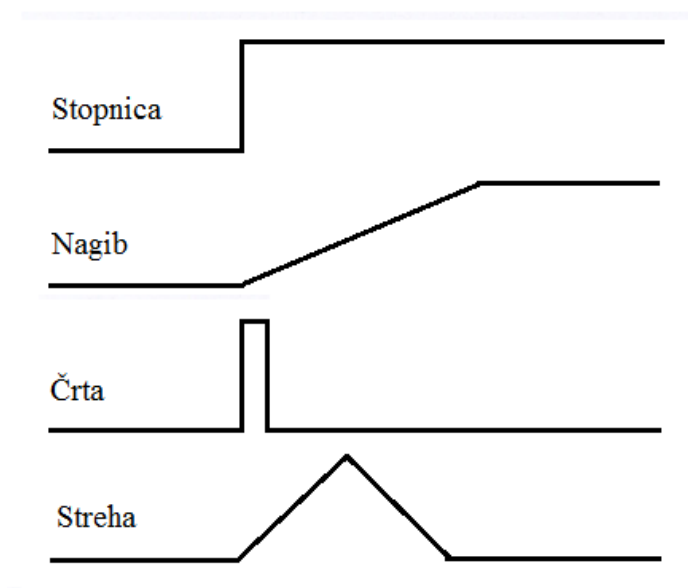
3.1 Iskanje robov

Rob je definiran kot znatna lokalna sprememba v intenziteti slike. Iskanje robov je tehnika, ki te spremembe poišče. Uporablja se za segmentacijo slik in pridobivanje podatkov na področjih, kot so strojni vid, računalniški vid in procesiranje slik. Poznamo vrsto algoritmov za detekcijo robov, kot so Robertsov, Sobelov, Cannyjev in Prewittov algoritem.

V idealnem primeru bi bil rezultat iskanja robov sklop povezanih krivulj, ki nakazujejo meje objektov. To pomeni, da uporaba tega algoritma na sliki znatno zmanjša količino podatkov, ki jih je treba procesirati, in lahko odstrani manj pomembne informacije, medtem ko ohranja pomembne strukturne lastnosti slike.

Seveda pa ni vedno mogoče z vsake slike izvleči idealne robove. Robovi pri nekoliko bolj zapletenih slikah so velikokrat razdrobljeni, kar pomeni, da krivulje niso povezane med seboj, prav tako pa lahko dobimo lažne robove, ki ne ustrezajo informacijam na sliki in tako dodatno otežijo branje ključnih podatkov s slike [16], [17].

Idealen rob je tak, kjer se sprememba intenzitete zgodi v koraku enega piksla. Takemu robu rečemo stopnica (angl. *step*), poznamo pa tudi črto (angl. *line*). Pri tej vrsti roba se vrednost, potem ko se je v koraku enega piksla spremenila, kmalu vrne nazaj na začetno vrednost. V dejanskih slikah pa sta ti dve vrsti roba zelo redki. Zaradi postopka glajenja slik, ki ga pozna večina naprav za zajem slik oziroma nizkofrekvenčnih komponent, se spremembe intenzitete redko zgodijo v koraku enega piksla. Zato se stopnica po navadi spremeni v tako imenovani nagib (angl. *ramp*), črta pa postane streha (angl. *roof*). Opisane vrste robov prikazuje slika 3.2.



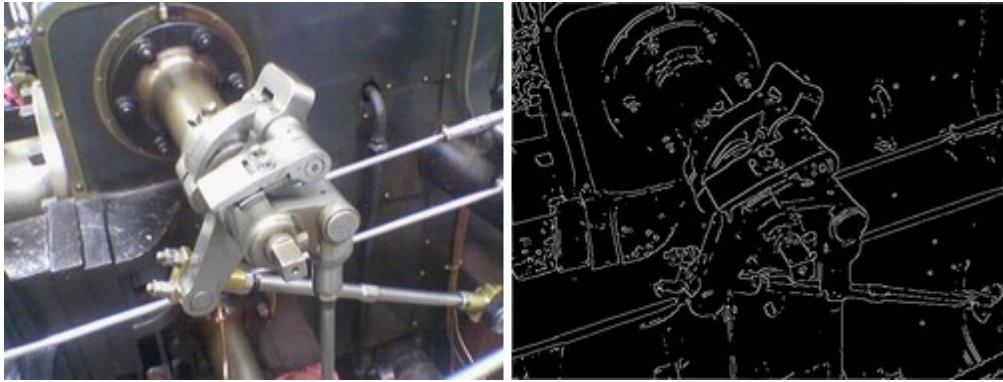
Slika 3.2: Vrste robov.

Iskanje robov po navadi poteka v treh fazah:

1. odstranitev šuma: iskanje robov nam največkrat oteži šum, zato skoraj vedno pred detekcijo robov izvedemo korak, ki se mu reče glajenje. Največkrat gre za Gaussov filter;
2. poudarjanje robov: filter prepozna ostre robove meja, ki v sliki ločujejo en predmet od drugega ali predmet od ozadja kontrastne barve. Nato poveča kontrast na območju neposredno okoli roba;
3. lociranje roba: v tem zadnjem koraku ugotovimo, kateri lokalni maksimumi so resnični robovi in kateri so posledica šuma.

3.1.1 Cannyjev algoritem

Cannyjev algoritem je od vseh algoritmov za iskanje robov najmanj občutljiv na šum, zato se ga najpogosteje uporablja. Algoritem drastično zmanjša količino podatkov, ki jih je treba obdelati. Primer delovanja algoritma prikazuje slika 3.3.



Slika 3.3: Levo: originalna slika, desno: slika po uporabi Cannyjevega algoritma.

Cannyjev algoritem je sestavljen iz več podalgoritmov. Kot prvi se izvede algoritem Canny Enhancer. Ta poskrbi za odstranjevanje šuma, zazna gradient in analizira moč roba. Če ne vemo, za kateri šum gre, se predpostavlja, da gre za Gaussov šum.

Predpostavimo, da imamo sliko I . Nad njo najprej izvedemo Gaussov filter, da se znebimo šuma.

$$J = I * g \quad (3.1)$$

Dobimo novo sliko J . Nato moramo nad vsako slikovno točko oziroma pikslom (x, y) v novi sliki J izvesti naslednje operacije:

1. izračunati gradientne komponente J_i in J_j ,
2. predvideti moč roba – E_s , pri čemer nam pomaga spodnja enačba

$$e_s(x,y) = \sqrt{J_i(x,y)^2 + J_j(x,y)^2},$$
3. predvideti smer normale na rob – E_o , pri čemer nam pomaga spodnja enačba

$$e_o(x,y) = \arctan\left(\frac{J_j}{J_i}\right).$$

Rezultat tega sta močnostna slika (angl. *strength image*) E_s , ki jo oblikujejo vrednosti $e_s(x,y)$, in orientacijska slika E_o , ki jo oblikujejo vrednosti $e_o(x,y)$.

Naslednji korak je izvedba algoritma dušenja nemaksimumov (angl. *Non Maxima Supression*) nad močnostno sliko E_s , upoštevajoč štiri smeri: 0° , 45° , 90° , 135° , in sicer zato, da odstranimo točke, ki niso maksimumi. Dobimo novo sliko I_n .

Za vsako slikovno točko (x,y) v močnostni sliki E_s moramo najti smer $d_k \in \{0^\circ, 45^\circ, 90^\circ, 135^\circ\}$, ki je najbližja izračunani vrednosti $e_o(x,y)$. Če velja, da je vrednost $E_s(x,y)$ manjša od vrednosti vsaj enega izmed njegovih sosedov ob normali d_k , potem velja, da je $I_n(x,y) = 0$, v nasprotnem

primeru pa $I_n(x,y) = e_s(x,y)$. Nova slika I_n je sestavljena iz točk, ki predstavljajo stanjšane robove. Če poenostavimo, gre torej za sliko E_s , potem ko ji odstranimo točke, ki niso maksimumi.

Ta slika pa še vedno vsebuje lokalne maksimume, ki so posledica šuma. Za odstranitev le-teh poskrbi zadnji korak v tem procesu, to je histerzni pragovni algoritem (angl. *Hysteresys Threshold Algorithm*). Ta algoritem uporablja vrednosti t_l in t_h , ki predstavljata minimalno in maksimalno pragovno vrednost. V prvem koraku lociramo naslednjo še neobiskano točko $I_n(x,y)$, za katero velja $I_n(x,y) > t_h$. Nato začne iz točke $I_n(x,y)$ in nadaljuje po povezani verigi lokalnih maksimumov v obe smeri, pravokotno na normalo roba, vse dokler velja $I_n(x,y) > t_l$. Vse vidne točke so nato označene, vrne pa nam tudi seznam lokacij vseh točk v povezanem obrisu. Končni rezultat je torej set seznamov, od katerih vsak opisuje povezan obris v sliki, prav tako pa tudi močnostna in orientacijska slika, ki opisujeta lastnosti robnih točk [13], [14], [15].

3.2 Odstranjevanje šuma

V digitalnih slikah se pogosto pojavljajo različne vrste šuma. Šum nastane zaradi napak pri zajemanju slik. Rezultat tega so vrednosti pikslov, ki ne odražajo prave intenzivnosti resničnega prizora. Gre torej za nihanje intenzitete. Šum vidimo kot posamezne piksele, ki so lahko svetlejši ali temnejši od okoliških pikslov. To so lokalni ekstremi. Svetlejši piksli so lokalni maksimumi, temnejši pa lokalni minimumi [18].

Poznamo več vrst šuma. Najpogostejša sta Gaussov šum ter šum sol in poper.

3.2.1 Gaussov šum

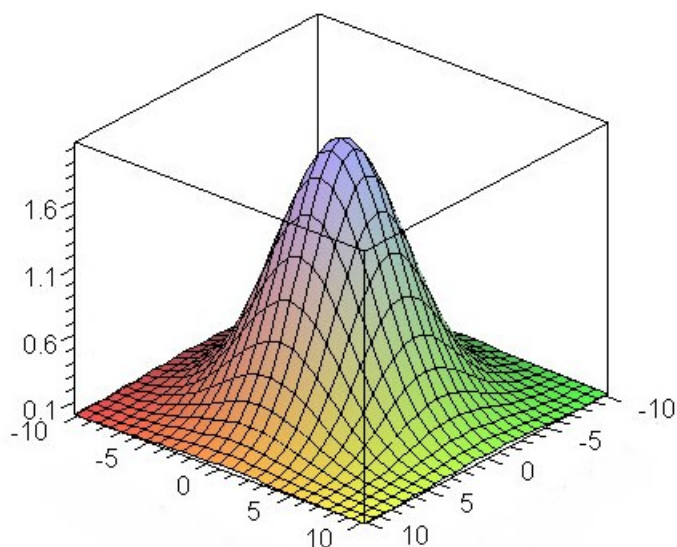
Gre za statističen šum, ki ima funkcijo verjetnostne porazdelitve, enake tisti normalne oziroma Gaussove porazdelitve. Glavni viri Gaussovega šuma v digitalnih slikah se pojavijo med pridobivanjem (lahko gre za šum senzorja, ki ga povzročata slaba osvetlitev ali visoka temperatura) ali med prenosom (šum elektronskega vezja) [19], [21].

Verjetnostno funkcijo lahko opišemo s formulo $p_G(z) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(z-\mu)^2}{2\sigma^2}}$, kjer z predstavlja nivo sivine, μ predstavlja srednjo vrednost in σ je standardni odklon.

Pri digitalni obdelavi slik lahko Gaussov šum zmanjšamo s prostorskim filtrom, vendar se lahko zgodi, da dobimo pri glajenju slik določene stranske učinke, kot na primer zabrisane robove in podrobnosti v sliki, ker te ustrezajo tudi blokiranim visokim frekvencam. Klasični prostorski filtri za odstranjevanje šuma vključujejo: filter povprečja (angl. *mean filter*), filter mediana (angl. *median filter*) in Gaussovo glajenje.

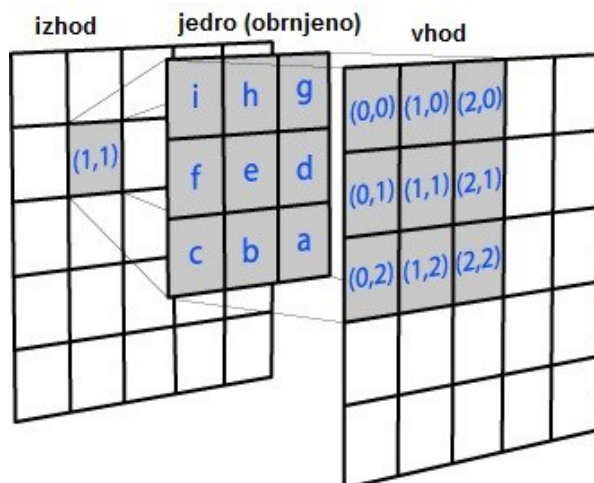
Gaussovo glajenje je nizkoprepustni filter, kar v praksi pomeni, da imamo neko določeno frekvenco, ki se ji reče odrezna frekvenca. Nizkoprepustni filter prepusti le signale s frekvenco, nižjo od odrezne frekvence.

Za postopek Gaussovega glajenja uporabimo tako imenovano Gaussovo jedro (angl. *kernel*), ki ni nič drugega kot polje pikslov – lahko bi rekli majhna slika. Vrednosti pikslov v tem polju so določene z dvodimenzionalno Gaussovo funkcijo $G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$, kot prikazuje graf na sliki 3.4 [19], [20].



Slika 3.4: Graf dvodimenzionalne Gaussove funkcije.

Vsak piksel na sliki se pomnoži z Gaussovim jedrom. Centralni piksel jedra, ki mu pred tem obrnemo vrstice in stolpce, se položi na piksel slike. Nato se pomnoži vrednosti z originalne slike s piksli v jedru, ki se prekrivajo. Te vrednosti se seštejejo in rezultat se uporabi za vrednost v ciljnim pikslu. Temu procesu pravimo konvolucija (slika 3.5).



Slika 3.5: Konvolucija.

Vsekakor pa bi bil v primeru slike, velikosti 1024×1024 , in Gaussovega jedra, velikosti 33×33 , ta proces prezahteven, saj bi potreboval $1024 \times 1024 \times 33 \times 33 = 1,14$ milijarde operacij, če hočemo izvesti filter nad celotno sliko. Zato lahko razdelimo dvodimenzionalno Gaussovo funkcijo v dve enodimenzionalni, kar pomeni, da namesto enega filtra $N \times N$, dobimo dva filtra $N \times 1$, in sicer horizontalni filter ter vertikalni filter. Vsak piksel nato pomnožimo z obema filtroma in seštejemo rezultate, da dobimo vrednost ciljnega piksla. Število operacij se zaradi tega drastično zmanjša.

3.3 Ostrenje slike

Ostrenje slik uporabimo, kadar želimo poudariti podrobnosti v sliki oziroma za zmanjševanje zamegljenosti, ne da bi pri tem nastal šum ali druge nepravilnosti. Gre torej za nasprotno operacijo zameglitve slike. Če smo pri odstranjevanju šuma oziroma zameglitvi uporabljali nizkoprepustni filter, moramo v tem primeru uporabiti visokoprepustni filter. Ta filter dela povsem enako kot nizkoprepustni, le da uporablja drugačno jedro za konvolucijo. Vendar ima ta filter slabo stran, saj poudari šum. Prav tako lahko preveč poudari nekatere majhne in blede podrobnosti, kar povzroči, da slika postane preveč zrnata in nenaravna. Filter torej lahko uporabimo za ostrenje, vendar lahko njegova prekomerna uporaba znatno zmanjša kakovost slike. Zaradi tega pogosto namesto navadnega jedra oziroma filtra uporabimo tako imenovano neostro masko (angl. *unsharp mask*) [22].

3.3.1 Neostra maska

Neostri filter je preprost operator za ostrenje, katerega ime izvira iz dejstva, da izboljša robove in druge visokofrekvenčne komponente na sliki skozi proces, ki odšteje neostro oziroma

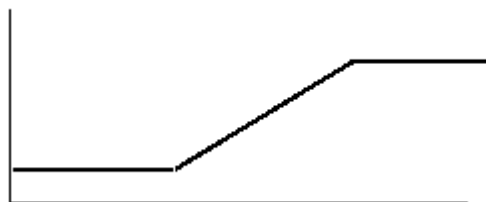
zamegljeno različico slike od originalne slike. Zelo pogosto se uporablja v programih za izboljšavo slik, tudi v programu Adobe Photoshop.

Neostro maskiranje ustvari robno sliko $g(x,y)$ iz vhodne slike $f(x,y)$ na način, kot prikazuje spodnja formula.

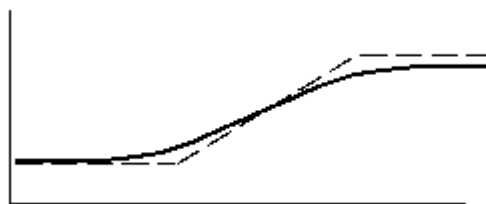
$$g(x,y) = f(x,y) - f_{smooth}(x,y), \quad (3.2)$$

kjer je $f_{smooth}(x,y)$ glajena različica originalne slike $f(x,y)$. Za glajenje uporabimo Gaussov filter.

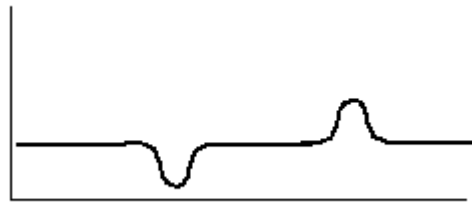
Operacijo izostritve z neostro masko lažje razumemo, če si pogledamo lastnosti njenega frekvenčnega odziva. Na sliki 3.6 imamo originalen signal. Če mu odštejemo njegovo nizkopasovno komponento, kot je prikazano na sliki 3.7, nam vrne visokopasovni oziroma robni prikaz (slika 3.8) [22].



Slika 3.6: Originalen signal.



Slika 3.7: Nizkopasovni signal.



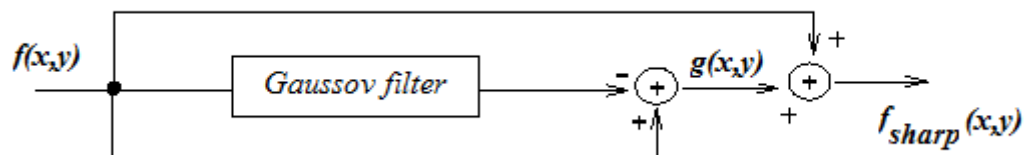
Slika 3.8: Visokopasovni signal.

Robno sliko lahko uporabimo za ostrenje, tako da jo dodamo originalni sliki (slika 3.9).



Slika 3.9: Izostren signal.

Celosten operator ostrenja z neostro masko je prikazan na sliki 3.10.

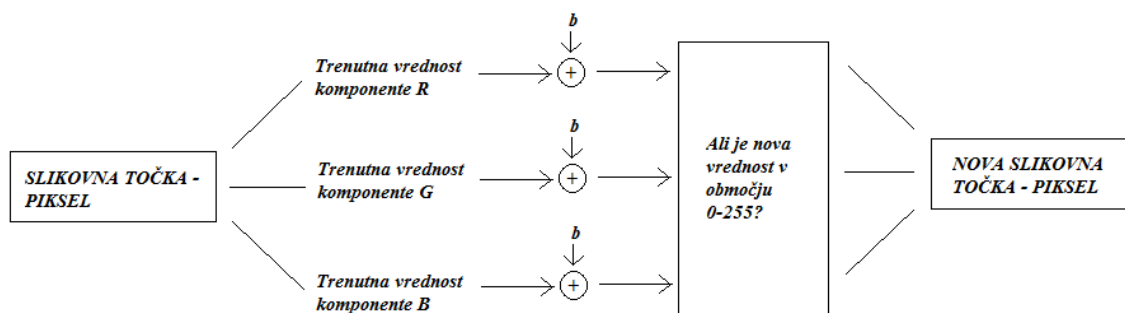


Slika 3.10: Operator ostrenja z neostro masko.

Vse skupaj lahko zapišemo z enačbo $f_{sharp}(x,y) = f(x,y) + k * g(x,y)$, kjer je k prilagoditvena konstanta. Realne vrednosti k se gibljejo med 0,2 in 0,7, kjer večje vrednosti pomenijo več izostritve [22], [23].

3.4 Spreminjanje svetlosti in kontrasta

Spreminjanje svetlosti je ena najlažjih operacij pri obdelavi slik, hkrati pa nam lahko zelo pomaga. Deluje tako, da vsem trem komponentam barvne sheme RGB pri vsakem izmed pikslov na sliki dodamo enako vrednost. Negativne vrednosti potemnijo sliko, medtem ko jo pozitivne posvetlijo. Vrednosti morajo biti v veljavnem območju sheme RGB, ki je 0–255 [24].



Slika 3.11: Posvetlitveni algoritem.

Slika 3.11 prikazuje delovanje algoritma za posvetlitev slike, pri čemer konstanta b predstavlja vrednost, ki jo dodamo vsaki izmed barvnih komponent RGB.

Pri spreminjanju kontrasta je postopek podoben. Najprej moramo izračunati faktor korekcije kontrasta F .

$$F = \frac{259(C+255)}{255(259-C)} \quad (3.3)$$

C predstavlja želeno stopnjo kontrasta. Vrednost mora biti v območju od -255 do 255 , kjer negativne vrednosti zmanjšajo količino kontrasta, pozitivne pa jo povečajo. Naslednji korak je izvajanje same korekcije kontrasta. Enako kot pri spreminjanju svetlosti moramo to storiti za vse tri barvne komponente sheme RGB. Pri tem je treba paziti, da je nova vrednost v območju od 0 do 255.

$$R' = F(R - 128) + 128 \quad (3.4)$$

R' predstavlja novo vrednost rdeče barve R . Enako storimo še za komponenti G in B , da dobimo novo barvo slikovne točke – piksla [25].

3.5 Segmentacija slik

Eden pomembnejših postopkov za odkrivanje ključnih informacij s slike je segmentacija. Poznamo več različnih algoritmov za segmentacijo: SLIC (angl. *Simple Linear Iterative Clustering*), SEEDS (angl. *Superpixels Extracted via Energy-Driven Sampling*) in Watershed. Segmentacija razdeli sliko na segmente – superpiksle, katerih namen je poenostaviti sliko v nekaj bolj razumljivega za lažjo analizo. Proces se največkrat uporablja za iskanje objektov in meja na sliki. Piksli so združeni v segmente glede na njihove lastnosti (barvo, teksturo ...). Segmentacijo lahko uporabljamo za iskanje podobnih slik po bazi slik – poizvedba glede na vsebino slike, iskanje obrazov in prstnih odtisov, v medicini – lociranje tumorjev, načrtovanje operacij.

3.6 Superresolucija

Superresolucija (angl. *super-resolution*) je ime za seznam metod, katerih skupni cilj je izboljšati prostorsko ločljivost slik oziroma videoposnetkov. Tipična področja uporabe so obdelava slik v medicini, daljinsko zaznavanje, radarski zajem slik, nadzorni videosistemi in računalniška forenzika.

Splošno gledano je ločljivost slik količina podrobnosti, ki jih vsebuje slika. Lahko jo opišemo na več načinov glede na to, s katerega vidika gledamo. Govorimo lahko o prostorski, časovni, spektralni, optični ločljivosti itd. Prostorska ločljivost opisuje nivo vizualnih podrobnosti, razvidnih s slike. Z drugimi besedami, oceni, kako blizu sta lahko dve črti na sliki, da sta še vedno razvidni. Ta definicija velja tako za analogne kot za digitalne kamere in slike. Pri digitalnih slikah, kjer je slikovna točka osnovna enota slike, obstaja povezava med prostorsko ločljivostjo in številom slikovnih točk. Število učinkovitih slikovnih točk, ki jih ima slikovni senzor oziroma digitalna kamera, je število osnovnih točkovnih senzorjev, ki pripevajo h končni sliki. Višja kot je ta številka, večja bo ločljivost in jasnejši bo predmet na sliki. V praksi po navadi upoštevamo samo skupno število slikovnih točk horizontalno in vertikalno, ki definirajo ločljivost (angl. *pixel resolution*) in hkrati služijo kot zgornja meja za prostorsko ločljivost. Ločljivost opišemo s setom dveh celih pozitivnih števil, kjer je prvo število število stolpcev slikovnih točk (širina), drugo pa število vrstic slikovnih točk (višina).

Superresolucijski algoritmi se razlikujejo po številu vhodnih in izhodnih slik, vključenih v proces. Ko z ene slike nizke ločljivosti (LR) dobimo eno sliko visoke ločljivosti (HR), govorimo o superresoluciji SISO (angl. *single-image, single-output*), ki se uporablja pri algoritmih za rekonstrukcijo slik ali za prepoznavo predmetov. Drugi superresolucijski algoritmi pa delujejo tako, da z integracijo več slik nizke ločljivosti ustvarijo sliko visoke ločljivosti. V tem primeru govorimo o superresoluciji MISO (angl. *multiple-image, single-output*). Slabost superresolucije

MISO je, da ne moremo izboljšati slik, ki vključujejo premikanje, oziroma subjekta, ki se premika. Deluje le na statičnih slikah. Tu gre za tako imenovane večokvirne (angl. *multi-frame*) metode, medtem ko pri SISO govorimo o enookvirnih (angl. *single-frame*) metodah. Primer praktične uporabe tega algoritma je prepoznavanje registrskih tablic z videoposnetka, da izboljšamo stopnjo alfanumerične prepoznave. Naslednji algoritem superresolucije je MIMO (angl. *multiple-image, multiple-output*), ki ga poznamo tudi pod imenom »video-to-video«. Ta algoritem poskuša rekonstruirati set slik visoke ločljivosti iz ekvivalentnega seta slik nizke ločljivosti. Tipičen primer uporabe algoritma MIMO je izboljšava kakovosti videoposnetkov iz videonadzornih kamer [26], [28].

3.6.1 Večokvirne superresolucijske metode

Osnovna predpostavka za izboljševanje prostorske ločljivosti pri večokvirnih (angl. *multi-frame*) superresolucijskih metodah je, da imamo na voljo več slik nizke ločljivosti istega prizora. Metode so najbolj učinkovite takrat, ko ima vsaka slika malo drugačen zorni kot. V tem primeru se vsaka slika obravnava kot nekoliko degradirana verzija visokoločljivostne slike. Težava superresolucije je po navadi modelirana kot inverzni problem, kjer mora biti izvorna informacija (visokoločljivostna slika) predvidena iz opazovanih podatkov – slik nizke ločljivosti. Reševanje inverzne težave po navadi zahteva najprej izgradnjo opazovalnega modela (angl. *observation model*). Večino naprav za zajem slik ali videoposnetkov lahko opišemo kot objektiv in zaslonka, ki ustvarita zamegljene slike, pokvarjene z nekaj dodatnega šuma. Če upoštevamo dano sliko visoke ločljivosti z N številom slikovnih točk (sliko lahko izrazimo kot vektor $x \in \mathbb{R}^N$), lahko opazovani model, ki opisuje zajem seta p slik nizke ločljivosti $y_k \in \mathbb{R}^M$, izrazimo kot:

$$y_k = DB_k W_k x + n_k \quad \text{za } 1 \leq k \leq p. \quad (3.5)$$

W_k je matrika $N \times N$, ki preslika koordinate slike visoke ločljivosti na nov sistem popačenih koordinat. B_k je zamegljena matrika $N \times N$, ki modelira različne pojave (npr. optični sistem in premike med procesom zajemanja), D je poenostavljena matrika $M \times N$ in n_k predstavlja urejen vektor šuma. Torej so opazovane nizkoločljivostne slike rezultat zameglitve, ukrivitve in operacij podvzorčenja, izvedenih nad isto visokoločljivostno sliko x . Če je podvzorčni faktor s (nizkoločljivostne slike se poenostavijo za faktor s v vsaki prostorski smeri), potem je dimenzijsko razmerje med vektorjem visokoločljivostne slike x in vektorji nizkoločljivostnih slik y_k enako

$$N = s^2 M.$$

Večokvirne superresolucijske metode so usmerjene k obračanju opazovanega modela. V ta namen je bilo v zadnjem desetletju predstavljenih več metod. Te lahko zaradi težav pri

ocenjevanju vseh potrebnih parametrov izdelajo sorazmerno dobre višje ločljivosti slik le za majhne faktorje povečav – tipično za faktorje, manjše od 2.

Splošno gledano lahko večokvirne superresolucijske algoritme razdelimo glede na tri glavne pristope:

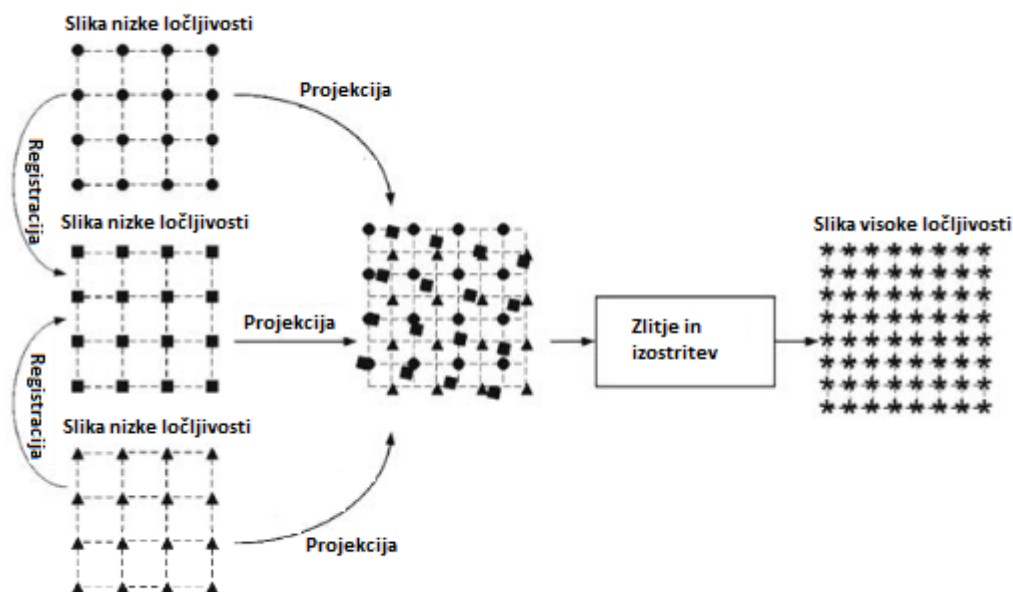
1. interpolacijski pristop,
2. frekvenčno-domenski pristop,
3. regularizacijski pristop.

V nadaljevanju naloge bomo pregledali te tri pristope, tako da bomo predstavili nekatere najbolj uporabljene metode [26].

3.6.1.1. Interpolacijski pristop

Interpolacijski superresolucijski algoritmi sestavijo sliko visoke ločljivosti tako, da projicirajo vse pridobljene slike nizke ločljivosti na referenčno sliko. Vse informacije, ki so na voljo z vsake slike, so nato zlite skupaj, saj vsaka nizkoločljivostna slika vsebuje količino dodatne informacije o prizoru. Na koncu nad sliko izvedemo izostritveni algoritem. Celoten pristop je prikazan na sliki 3.12 in je tipično sestavljen iz naslednjih treh korakov:

1. registracijski korak: za poravnavo vseh vhodnih nizkoločljivostnih slik,
2. interpolacijski korak: za izdelavo slike z višjo ločljivostjo,
3. izostritveni korak: za izostritev rekonstruirane visokoločljivostne slike.



Slika 3.12: Shema interpolacijskega pristopa večokvirne superresolucije.

Ključno vlogo v tem procesu igra interpolacija. Lahko jo izvedemo na več načinov. Najenostavnejši interpolacijski algoritem je algoritem najbližjega sosedu, kjer je vsaki neznani slikovni točki dodeljena vrednost intenzitete, ki je enaka njenim sosedom. Vendar pa ta metoda včasih ustvari slike, ki so videti preveč kockaste. Drugi tradicionalni pristopi vključujejo bilinearno in bikubično interpolacijo [26].

3.6.1.2 Frekvenčno-domenski pristop

Velik razred večokvirnih superresolucijskih metod uporablja frekvenčno-domensko formulacijo superresolucijske težave. Glavni princip je, da so namigi o visokih frekvencah porazdeljeni med več nizkoločljivostnih slik v obliki povezanih spektralnih frekvenc. Prva tovrstna metoda deluje tako, da najprej podatke slik nizke ločljivosti pretvori v domeno diskretne Fourierjeve transformacije in jih nato združi glede na razmerje med povezanimi koeficienti DFT opazovanih slik nizke ločljivosti. Pristop temelji na sledečih principih:

1. lastnost premika Fourierjeve transformacije,
2. povezovalno razmerje med neprekinjeno Fourierjevo transformacijo (CFT) in diskretno Fourierjevo transformacijo (DFT) opazovanih slik nizke ločljivosti,
3. domneva, da je originalna slika visoke ločljivosti pasovno omejena.

Te lastnosti nam omogočijo, da formuliramo enačbo, ki povezuje povezane koeficiente DFT opazovanih slik nizke ločljivosti z vzorcem neznane slike CFT visoke ločljivosti.

Frekvenčno domenski pristop k superresoluciji ima veliko prednosti. Prva je teoretična preprostost: razmerje med nizkoločljivostnimi vhodnimi slikami in viskoločljivostno sliko je jasno prikazano. Ta pristop torej predstavlja intuitiven način za izboljšavo podrobnosti v sliki, tako da ekstrapolira visokofrekvenčne informacije, prisotne na slikah nizke ločljivosti. Druga prednost je ta, da imajo ti pristopi nizko računsko kompleksnost, tako da so primerni tudi za paralelne implementacije.

V zadnjem času je več raziskovalcev začelo z raziskovanjem uporabe valovne transformacije za reševanje problema superresolucije, da obnovi podrobne informacije (po navadi so to visokofrekvenčne informacije), ki so izgubljene ali degradirane med postopkom zajemanja slike. Valovna transformacija zagotavlja močno in učinkovito večnivojsko reprezentacijo slike za obnavljanje visokofrekvenčnih informacij. Ti pristopi po navadi obravnavajo slike nizke ločljivosti kot nizkoprepustne filtrirane podpasove neznane valovno transformirane viskoločljivostne slike. Cilj je, da ocenimo podpasovne koeficiente, nato pa izvedemo inverzno valovno transformacijo, da dobimo sliko visoke ločljivosti. Slike nizke ločljivosti so obravnavane kot reprezentacija valovnih koeficientov po nivojih N dekompozicije. Nato se lahko slika visoke ločljivosti ustvari tako, da se oceni $(N+1)$ -te valovne koeficiente in izvede inverzno valovno dekompozicijo [26].

3.6.1.3 Regularizacijski pristop

Ker je superresolucijski izračun zaradi nezadostnega števila slik nizke ločljivosti ali slabo pogojenih zameglitvenih operatorjev v bistvu slabo zastavljen inverzni problem, so bili številni regularizacijski superresolucijski algoritmi razviti tako, da stabilizirajo operacijo inverzije, s tem da omejijo število možnih rezultatov. Osnovna ideja teh pristopov je, da uporabljajo regularizacijsko strategijo, da vključijo neko predhodno znanje slike visoke ločljivosti. Metode lahko razdelimo v dve kategoriji:

1. deterministični regularizacijski pristopi,
2. stohastični regularizacijski pristopi.

Deterministični pristopi temeljijo na dejstvu, da s približki registracijskih parametrov lahko povsem določimo opazovalni model. Deterministični regularizacijski pristop nato reši inverzni problem, tako da izkoristi določene predhodne informacije o rešitvi, ki se lahko uporabijo, da naredijo problem dobro zastavljen. Tipična metoda se nanaša na rešitev problema omejenih najmanjših kvadratov – CLS. Ta problem lahko formuliramo tako, da izberemo tak x , da zmanjšamo Lagrangeovo funkcijo v skladu z naslednjo obliko:

$$\hat{\mathbf{x}} = \arg \min_{\mathbf{x}} \left\{ \sum_{k=1}^p \|\mathbf{y}_k - \mathbf{A}_k \mathbf{x}\|^2 + \alpha \|\mathbf{C} \mathbf{x}\|^2 \right\}, \quad (3.6)$$

kjer je \mathbf{A}_k globalna matrika, ki upošteva vse degradacijske operacije na okvirju \mathbf{y}_k , α je Lagrangeov multiplikator, \mathbf{C} pa je visokoprepustni filter. V zgornji formuli je apriori znanje, ki zadeva želeno rešitev, predstavljeno z zglajevalno omejitvijo. To pomeni, da se predpostavlja, da je večina slike že po naravi gladka, zato je primerno, da se zmanjša količina visokoprepustne energije v obnovljeni sliki. Lagrangeov multiplikator nadzira kompromis med zvestobo podatkom in zglajenostjo rezultata. Funkcionalni strošek v zgornji enačbi je konveksen in odvedljiv z uporabo kvadratnega regularizacijskega izraza. Torej lahko dobimo edinstveno oceno $\hat{\mathbf{x}}$, ki ga zmanjša. Ena osnovna deterministična iterativna tehnika vključuje reševanje:

$$\left(\sum_{k=1}^p \mathbf{A}_k^T \mathbf{A}_k + \alpha \mathbf{C}^T \mathbf{C} \right) \hat{\mathbf{x}} = \sum_{k=1}^p \mathbf{A}_k^T \mathbf{y}_k, \quad (3.7)$$

kar vodi do sledeče iterative za $\hat{\mathbf{x}}$:

$$\hat{\mathbf{x}}^{n+1} = \hat{\mathbf{x}}^n + \beta \left(\sum_{k=1}^p \mathbf{A}_k^T (\mathbf{y}_k - \mathbf{A}_k \hat{\mathbf{x}}^n) - \alpha \mathbf{C}^T \mathbf{C} \hat{\mathbf{x}}^n \right), \quad (3.8)$$

kjer β predstavlja konvergenčni parameter.

Stohastične regularizacijske metode po navadi uporabijo Bayesov pristop, po katerem so informacije, ki jih lahko izvlečemo iz opazovanj (slik nizke ločljivosti) o neznanem signalu (slik visoke ločljivosti), vsebovane v verjetnostni porazdelitvi neznanega. Te metode se nato zanašajo na verjetnost, da je posteriori funkcija gostote verjetnosti (PDF) originalne slike visoke ločljivosti. Ta je lahko vzpostavljena tako, da izkoristimo informacije, dane s strani opazovanih slik nizke ločljivosti in predhodnega znanja o sliki visoke ločljivosti.

Eden najbolj razširjenih Bayesovih superresolucijskih pristopov je maksimum a posteriori (MAP) ocenjevalni pristop. Ocenjevalnik MAP x poveča posteriori PDF $P(x|\mathbf{y}_k)$ glede na x :

$$\hat{\mathbf{x}} = \arg \max_{\mathbf{x}} P(\mathbf{x} | \mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_p) \quad (3.9)$$

Če vzamemo logaritmčno funkcijo in izvedemo Bayesov teorem nad pogojno verjetnostjo, lahko optimizacijo MAP zapišemo na sledeči način:

$$\hat{\mathbf{x}} = \arg \max_{\mathbf{x}} \{ \ln P(y_1, y_2, \dots, y_p | \mathbf{x}) + \ln P(\mathbf{x}) \} \quad (3.10)$$

Tako apriori model slike $P(x)$ kot pogojna gostota $P(y_1, y_2, \dots, y_p | x)$ sta definirana z apriori znanjem, ki zadeva sliko visoke ločljivosti x in statistično informacijo o šumu [26].

3.6.2 Enoslikovne superresolucijske metode

Pri teh metodah gre za sestavljanje slike visoke ločljivosti iz samo ene opazovane slike nizke ločljivosti. Začetna točka je torej ena sama slika nizke ločljivosti, ki smo jo pridobili v znanem ali neznanem postopku zajema slik. Proces oblikovanja slike nizke ločljivosti iz originalne slike visoke ločljivosti lahko zapišemo kot:

$$x = DHy + v, \quad (3.11)$$

kjer je x nizkoločljivostna slika in y viskoločljivostna slika, D predstavlja operacijo poslabšanja slike (angl. *downsampling*) za faktor s , H je zameglitveni operator, ki se po navadi modelira po Gaussu, v pa je aditivni šum. Dobimo torej zamegljeno in poslabšano sliko visoke ločljivosti.

Problem, predstavljen v zgornji enačbi, je sam po sebi slabo zastavljen, ker imamo lahko več slik visoke ločljivosti, ki generirajo enako sliko nizke ločljivosti. Enoslikovna superresolucija je tesno povezana s tradicionalno analitično intepolacijo, saj si delita isti cilj. Z računanjem manjkajočih slikovnih točk iz povprečja znanih slikovnih točk na sliki visoke ločljivosti dajejo tradicionalne interpolacijske metode prednost glajenju. Vendar pa naravne slike pogosto vsebujejo močne prekinitve, kot na primer robove in kote, zaradi česar so rezultat glajenja zamegljeni artefakti na izhodni sliki. Cilj superresolucije je torej ustvariti boljše rezultate.

Enoslikovne superresolucijske metode lahko razdelimo v dve kategoriji:

1. metode, ki temeljijo na učenju: uporabljajo metode strojnega učenja in pogosto ustvarijo slovar iz baze slik,
2. metode, ki temeljijo na rekonstrukciji: definirajo omejitve za ciljno sliko visoke ločljivosti, da izboljšajo kakovost rekonstrukcije.

Enoslikovna superresolucija, ki temelji na rekonstrukciji, vključuje različne metode. Več algoritmov te vrste so metode, ki se osredotočajo na robove, kar pomeni, da skušajo rekonstruirati podrobnosti slike z interpolacijo vhodne slike nizke ločljivosti, medtem ko se osredotočajo na ostrenje robov. Nekateri pristopi delujejo tako, da izvečejo robove vhodne slike, zato da se doseže njihovo kontinuiteto. Nato se jih zlije skupaj z rezultati interpolacije,

da dobimo končno sliko. Poznamo pa tudi pristop, ki poskuša reševati slabo zastavljen problem superresolucije skozi regularizacijske metode.

Značilna funkcija metod, ki temeljijo na učenju, je vključevanje tehnik strojnega učenja, da lokalno ocenimo visokoločljivostne podrobnosti izhodne slike. Ti algoritmi lahko obstajajo v postopkih, ki temeljijo na slikovnih točkah (angl. *pixel-based procedures*), kjer je vsaka vrednost v izhodni visokoločljivostni sliki močno povzeta iz statističnega učenja. Lahko pa obstajajo tudi v postopkih, ki temeljijo na območjih (angl. *patch-based procedures*), kjer je visokoločljivostna ocena izvedena, zahvaljujoč slovarju korespondenc nizko- in visokoločljivostnih območij – kvadratnih blokov slikovnih točk. Slovar, ustvarjen na način, da povezuje nizkoločljivostna območja z visokoločljivostnimi, je nato uporabljen nad dano sliko nizke ločljivosti, da obnovi njeno najbolj verjetno visokoločljivostno različico. Seveda je kakovost te obnovljene slike odvisna od kakovosti slovarja, kar pomeni, da rekonstrukcija resničnih in neznanih podrobnosti ni zagotovljena.

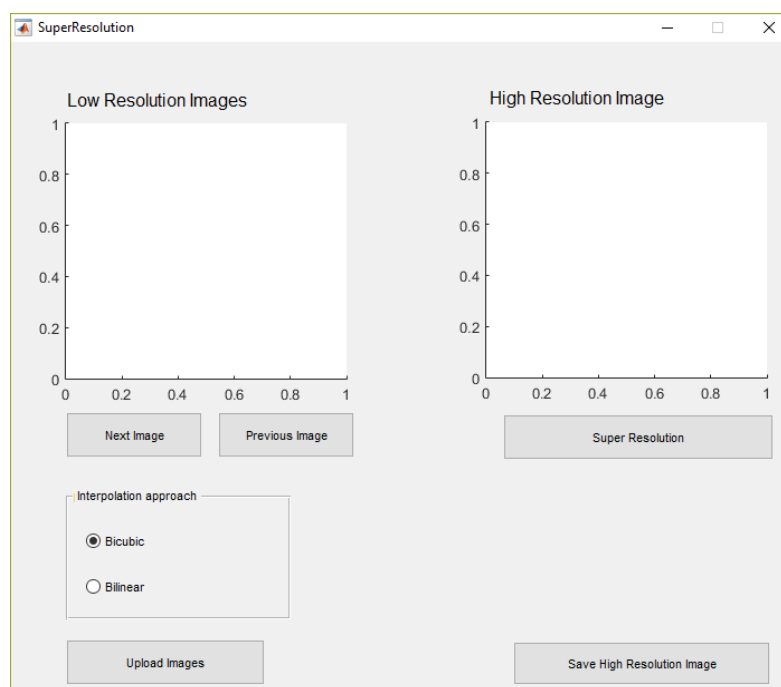
Enoslikovni superresoluciji, ki temelji na učenju in uporablja območja, se včasih reče tudi superresolucija, ki temelji na primerih (angl. *example-based*). V postopku povečave (angl. *upscaling*) se slika nizke ločljivosti razdeli na območja. Za vsako to območje se rekonstruira novo območje visoke ločljivosti z opazovanjem »primerov«, vsebovanih v slovarju.

Naslednja družina enoslikovnih superresolucijskih metod so metode, ki temeljijo na vdelavi sosedov (angl. *neighbor embedding*). Za te metode je značilna selekcija več nizkoločljivostnih kandidatov območij v slovarju po načinu najbližjega sosedu. Izhodna visokoločljivostna območja so nato rekonstruirana s kombiniranjem visokoločljivostnih verzij teh izbranih območij.

Poglavje 4 Implementacija superresolucije in rezultati

V aplikaciji smo implementirali interpolacijski pristop superresolucije. Aplikacija je napisana v programskem jeziku Matlab [27]. Uporabili smo bilinearni in bikubični algoritem. Algoritma najbližjega sosedra nismo implementirali, saj ta daje slabše rezultate (kockaste slike).

Kot vidimo na sliki 4.1, nam grafični vmesnik omogoča nalaganje nizkoločljivostnih slik. Idealno je, če ima vsaka slika malenkost drugačen zorni kot. Po nalaganju slik izberemo, kateri interpolacijski pristop želimo uporabiti. Po kliku na gumb Super-resolution nam program na desni strani prikaže rezultat.



Slika 4.1: Grafični vmesnik.

Da bi izvedeli, koliko nam superresolucija dejansko lahko pomaga pri izboljšanju slik, smo za primerjavo vzeli eno visokoločljivostno sliko I_1 . Nato smo vzeli več podobnih visokoločljivostnih slik (za prvi primer štiri slike, za drugi pa šest slik), izmed katerih je bila vsaka zajeta pod malenkost drugačnim zornim kotom. Pomanjšali smo jih za faktor 3, da smo dobili nizkoločljivostne slike. Primere originalnih slik visoke ločljivosti in slik nizke ločljivosti lahko vidimo na slikah 4.2 in 4.3.



Slika 4.2: Levo: originalna slika visoke ločljivosti I_1 , desno: štiri slike nizke ločljivosti, vir: <http://www.cns.nyu.edu/~lcv/iclr2017/>



Slika 4.3: Levo: originalna slika visoke ločljivosti I_1 , desno: šest slik nizke ločljivosti, vir: <https://www.bitnik.org/s/>

Pred samim postopkom interpolacije moramo v programu najprej izračunati, kam v novi visokoločljivostni sliki bodo padle že obstoječe slikovne točke, katerih vrednosti poznamo. Postopek poteka, tako da vzamemo prvo izmed nizkoločljivostnih slik za referenčno sliko. Nato pogledamo iste slikovne točke v vsaki naslednji sliki. Preverimo, ali se slikovni točki popolnoma ujemata ali obstaja razlika med njima. Če sta enaki, ju zavržemo, v nasprotnem primeru pa bomo uporabili vrednost te slikovne točke v novi visokoločljivostni sliki. Te vrednosti torej ni treba interpolirati, ker smo jo uporabili neposredno iz nizkoločljivostne slike.

Manjkajoče vrednosti v novi sliki nato izračunamo z bikubično ali bilinearno interpolacijo. Celoten postopek interpolacije se izvede v funkciji *griddata*, katere namen je, da interpolira razpršene podatke. Dobimo visokoločljivostno sliko I_2 .

Funkciji *griddata* podamo naslednje parametre:

- x – vektor, ki določa koordinate x vzorčnih točk,
- y – vektor, ki določa koordinate y vzorčnih točk,
- v – vektor vzorčnih vrednosti, ki ustrezajo vzorčnim koordinatam x in y ,
- xq – polje, ki določa koordinate x točk, po katerih poizvedujemo, in jih je treba izračunati,
- yq – polje, ki določa koordinate y točk, po katerih poizvedujemo, in jih je treba izračunati,
- *method* – parameter, ki določa, po kateri metodi (bikubični ali bilinearni) bomo izvedli interpolacijo.

Metoda nam vrne interpolirane vrednosti poizvedovanih točk. Pri dvodimenzionalni interpolaciji, kjer xq in yq predstavljata $m \times n$ mrežo točk, dobimo kot rezultat polje $m \times n$. Postopek interpolacije poteka za vsak barvni kanal posebej, zato moramo vse tri barvne kanale na koncu združiti v slikovno točko.

4.1 Rezultati bikubičnega interpolacijskega algoritma

Bikubična interpolacija se uporablja za interpolacijo točk na dvodimenzionalni mreži. Interpolirana površina je bolj zglajena kot pri bilinearni interpolaciji ali interpolaciji najbližjega soseda. Pri obdelavi slik je ta algoritem bolj v uporabi kot prejšnja dva, in sicer takrat, kadar nam hitrost ni pomembna. V nasprotju z bilinearno interpolacijo, ki upošteva le 4 (2×2) slikovne točke, jih bikubična interpolacija upošteva 16 (4×4). Slike, nad katerimi je bil izveden ta algoritem, so bolj gladke in imajo manj interpolacijskih nepravilnosti.

Predpostavimo, da so vrednosti funkcije f in štirih odvodov f_x , f_y in f_{xy} znane na štirih kotih $(0, 0)$, $(1, 0)$, $(0, 1)$ in $(1, 1)$. Interpolirano površino lahko nato zapišemo kot:

$$p(x, y) = \sum_{i=0}^3 \sum_{j=0}^3 a_{ij} x^i y^j \quad (4.1)$$

Težava interpolacije vključuje določitev 16 koeficientov a_{ij} . Uskladitev $p(x, y)$ z vrednostmi funkcije nam vrne naslednje štiri enačbe:

$$f(0, 0) = p(0, 0) = a_{00}$$

$$\begin{aligned}
f(1,0) &= p(1,0) = a_{00} + a_{10} + a_{20} + a_{30} \\
f(0,1) &= p(0,1) = a_{00} + a_{01} + a_{02} + a_{03} \\
f(1,1) &= p(1,1) = \sum_{i=0}^3 \sum_{j=0}^3 a_{ij}
\end{aligned} \tag{4.2}$$

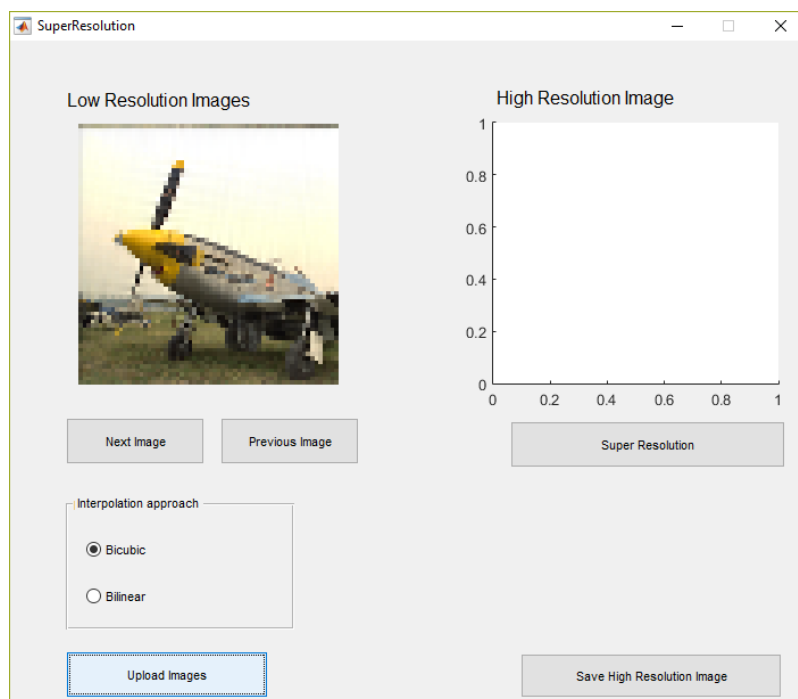
Dobimo osem enačb za odvode v smeri x in y :

$$\begin{aligned}
f_x(0,0) &= p_x(0,0) = a_{10} \\
f_x(1,0) &= p_x(1,0) = a_{10} + 2a_{20} + 3a_{30} \\
f_x(0,1) &= p_x(0,1) = a_{10} + a_{11} + a_{12} + a_{13} \\
f_x(1,1) &= p_x(1,1) = \sum_{i=1}^3 \sum_{j=0}^3 a_{ij} i \\
f_y(0,0) &= p_y(0,0) = a_{01} \\
f_y(1,0) &= p_y(1,0) = a_{01} + a_{11} + a_{21} + a_{31} \\
f_y(0,1) &= p_y(0,1) = a_{01} + 2a_{02} + 3a_{03} \\
f_y(1,1) &= p_y(1,1) = \sum_{i=0}^3 \sum_{j=1}^3 a_{ij} j
\end{aligned} \tag{4.3}$$

Štiri enačbe za odvod xy :

$$\begin{aligned}
f_{xy}(0,0) &= p_{xy}(0,0) = a_{11} \\
f_{xy}(1,0) &= p_{xy}(1,0) = a_{11} + 2a_{21} + 3a_{31} \\
f_{xy}(0,1) &= p_{xy}(0,1) = a_{11} + 2a_{12} + 3a_{13} \\
f_{xy}(1,1) &= p_{xy}(1,1) = \sum_{i=1}^3 \sum_{j=1}^3 a_{ij} ij
\end{aligned} \tag{4.4}$$

Ta postopek vrne površino $p(x,y)$ na kvadratni enoti $[0,1] \times [0,1]$, ki je neprekinjena in z neprekinjenimi odvodi. Bikubično interpolacijo na poljubno veliki mreži lahko dosežemo z lepljenjem takih bikubičnih površin skupaj, pri čemer moramo paziti, da se na mejah površin odvodi ujemajo [29].



Slika 4.4: Nalaganje slik nizke ločljivosti.

Kot vidimo na sliki 4.4, s klikom na gumb Upload images izberemo mapo, v kateri so slike nizke ločljivosti, in jih naložimo v program. Z gumboma Previous image in Next image se lahko pomikamo med naloženimi slikami nizke ločljivosti. Bikubično interpolacijsko metodo izberemo tako, da pod gumbom Interpolation approach izberemo Bicubic in nato kliknemo na gumb Super-resolution. Dobimo rezultat, kot ga vidimo na slikah 4.5 in 4.6. S klikom na gumb Save high resolution image lahko novo nastalo sliko tudi shranimo.

Slika 4.5: Levo: visokoločljivostna slika I_2 , ustvarjena z bikubično interpolacijo, desno: originalna visokoločljivostna slika I_1 .



Slika 4.6: Levo: visokoločljivostna slika I_2 , ustvarjena z bikubično interpolacijo, desno: originalna visokoločljivostna slika I_1 .

V prvem primeru smo uporabili štiri slike, v drugem pa šest. Boljši rezultat bi lahko dobili, če bi uporabili več slik nizke ločljivosti, seveda ob predpostavki, da je med vsako sliko manjša razlika (rotacija, zamik). Na ta način dobimo bolj natančne informacije o dejanski sliki in bolj natančno lahko sestavimo končno sliko. Več slikovnih točk kot pridobimo s tem postopkom, manj dela preostane za postopek interpolacije. Večje število slik nam ne pomaga v primeru, da so vse slike identične, saj nam vsaka slika vrne slikovne točke enake vrednosti.

Da si lažje predstavljamo, kako pomembno je za doseganje čim boljših rezultatov uporabiti več nizkoločljivostnih slik, imamo na sliki 4.7 primer visokoločljivostne slike, ustvarjene z bikubično interpolacijo na podlagi ene same nizkoločljivostne slike.



Slika 4.7: Levo: slika I_2 , ustvarjena z bikubično interpolacijo na podlagi več nizkoločljivostnih slik, desno: slika, ustvarjena z bikubično interpolacijo na podlagi ene nizkoločljivostne slike.

4.2 Rezultati bilinearnega interpolacijskega algoritma

Ko sliko povečujemo, se mora vsaka slikovna točka originalne slike premakniti v določeno smer glede na povečevalno konstanto. Kadar povečujemo sliko za neintegralen povečevalni faktor, nastanejo slikovne točke oziroma tako imenovane luknje, katerim niso dodeljene primerne vrednosti. V tem primeru jim moramo dodeliti primerne RGB oziroma sivinske vrednosti, tako da končna slika ne vsebuje slikovnih točk brez vrednosti.

Bilinearno interpolacijo lahko uporabimo takrat, kadar popolna transformacija slike z ujemanjem slikovnih točk ni mogoča, tako da lahko izračunamo in dodelimo primerne intenzitetne vrednosti slikovnim točkam. Za razliko od drugih interpolacijskih tehnik, kot sta interpolacija najbližjega sosedu in bikubična interpolacija, bilinearna interpolacija uporabi samo štiri najbližje vrednosti slikovnih točk, ki so postavljene v diagonalni smeri od omenjene slikovne točke, da poišče primerno vrednost tej točki. Bilinearna interpolacija upošteva najbližjo 2×2 soseseo znanih slikovnih točk, ki obkroža lokacijo slikovne točke z neznano vrednostjo. Nato vzame obteženo povprečje teh štirih slikovnih točk, da dobi končno interpolirano vrednost. Teža vsake izmed štirih točk je določena glede na izračunano razdaljo slikovne točke od vsake od znanih točk v dvodimenzionalnem prostoru.

Predpostavimo, da želimo dobiti vrednost neznane funkcije f na točki (x, y) , upoštevajoč, da poznamo vrednost f na štirih točkah $Q_{11} = (x_1, y_1)$, $Q_{12} = (x_1, y_2)$, $Q_{21} = (x_2, y_1)$ in $Q_{22} = (x_2, y_2)$. Naprej izvedemo linearno interpolacijo v smeri x . Dobimo:

$$\begin{aligned} f(x, y_1) &\approx \frac{x_2 - x}{x_2 - x_1} f(Q_{11}) + \frac{x - x_1}{x_2 - x_1} f(Q_{21}) \\ f(x, y_2) &\approx \frac{x_2 - x}{x_2 - x_1} f(Q_{12}) + \frac{x - x_1}{x_2 - x_1} f(Q_{22}) \end{aligned} \quad (4.5)$$

Nadaljujemo z interpolacijo v smeri y , da dobimo željeno oceno.

$$\begin{aligned} f(x, y) &\approx \frac{y_2 - y}{y_2 - y_1} f(x, y_1) + \frac{y - y_1}{y_2 - y_1} f(x, y_2) \\ &\approx \frac{y_2 - y}{y_2 - y_1} \left(\frac{x_2 - x}{x_2 - x_1} f(Q_{11}) + \frac{x - x_1}{x_2 - x_1} f(Q_{21}) \right) \\ &\quad + \frac{y - y_1}{y_2 - y_1} \left(\frac{x_2 - x}{x_2 - x_1} f(Q_{12}) + \frac{x - x_1}{x_2 - x_1} f(Q_{22}) \right) \\ &\approx \frac{1}{(x_2 - x_1)(y_2 - y_1)} [x_2 - x \quad x - x_1] \begin{bmatrix} f(Q_{11}) & f(Q_{12}) \\ f(Q_{21}) & f(Q_{22}) \end{bmatrix} \begin{bmatrix} y_2 - y \\ y - y_1 \end{bmatrix} \end{aligned} \quad (4.6)$$

Enak rezultat dobimo, če interpoliramo najprej po smeri y in nato po smeri x [29].

Če v grafičnem vmesniku izberemo možnost Bilinear, dobimo rezultat, kot ga vidimo na slikah 4.8 in 4.9. Vidimo, da je rezultat skoraj identičen rezultatu bikubičnega interpolacijskega algoritma, a vendar je bikubični algoritem bolj natančen.

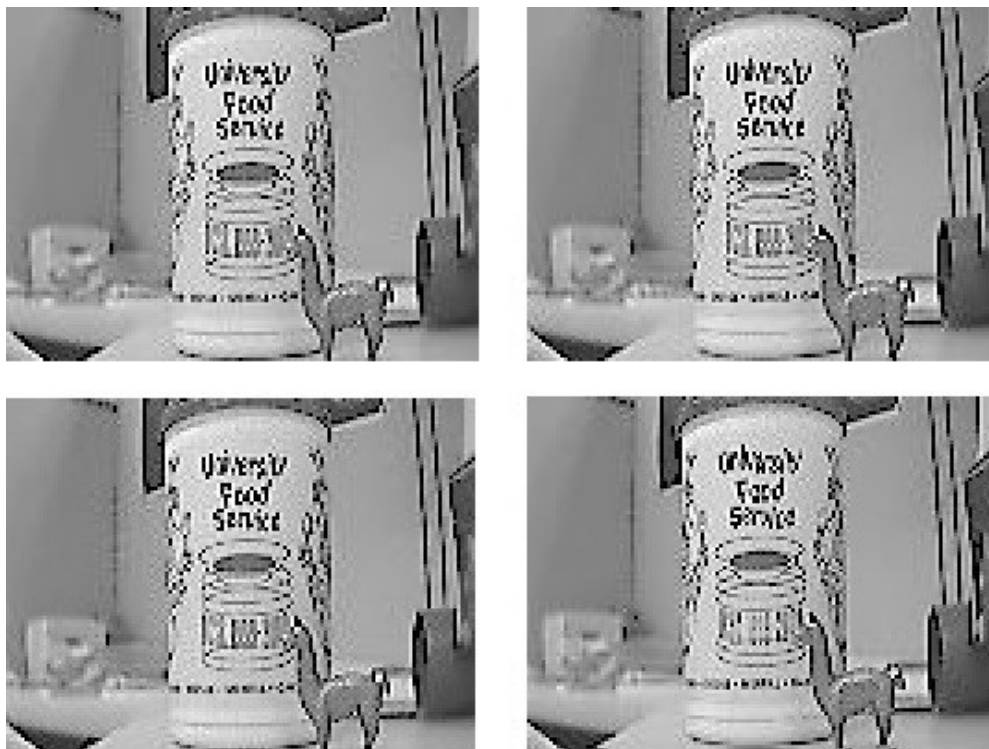


Slika 4.8: Levo: visokoločljivostna slika I_2 , ustvarjena z bilinearno interpolacijo, desno: originalna visokoločljivostna slika I_1 .



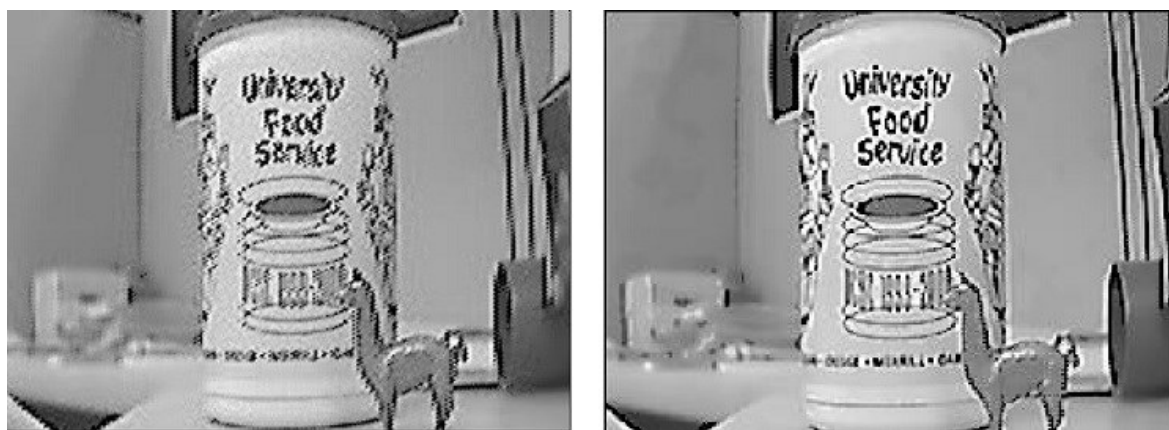
Slika 4.9: Levo: visokoločljivostna slika I_2 , ustvarjena z bilinearno interpolacijo, desno: originalna visokoločljivostna slika I_1 .

Pogledali si bomo še en primer z bilinearno interpolacijo na sliki z nekoliko več podrobnosti. Zanima nas, kakšna razlika nastane, če uporabimo pet nizkoločljivostnih slik in petdeset nizkoločljivostnih slik. Vse skupaj imamo torej petdeset nizkoločljivostnih slik. Na sliki 4.10 vidimo primere štirih slik. Razlika med slikami je tu nekoliko bolj vidna, še posebej če pogledamo napis, kar je pomembno za naš postopek, saj želimo pridobiti čim več informacij, ki nam bodo pomagale sestaviti končno sliko.



Slika 4.10: Primeri nizkoločljivostnih slik, vir: <http://www.cns.nyu.edu/~lcv/iclr2017/>.

Najprej bomo naredili superresolucijo s petimi nizkoločljivostnimi slikami in nato še s petdesetimi nizkoločljivostnimi slikami, in sicer s postopkom bilinearne interpolacije. Rezultat vidimo na sliki 4.11.



Slika 4.11: Levo: slika, ustvarjena na podlagi petih nizkoločljivostnih slik, desno: slika, ustvarjena na podlagi petdesetih nizkoločljivostnih slik.

Poglavje 5 Zaključek

Primerjali smo bilinearno in bikubično interpolacijsko metodo superresolucije. Metodi vsekakor dajeta obetavne rezultate, seveda pod ustreznimi pogoji, včasih pa to še vedno ni dovolj, da bi s slike razbrali ključno informacijo. Boljše rezultate smo dobili z bikubično metodo, ki pa je časovno bolj zahtevna. Opazili smo, da več nizkoločljivostnih slik kot vstavimo v program, boljše rezultate dobimo.

Razvoj metod superresolucije gre seveda naprej in pri podjetju Google že razvijajo nadgrajeno metodo superresolucije, ki so jo poimenovali RAISR (*Rapid and Accurate Image Super-Resolution*). RAISR vrača rezultate, ki so primerljivi oziroma v večini primerov boljši od rezultatov trenutnih metod superresolucije. To stori od 10- do 100-krat hitreje, tako da bi metoda lahko brez težav delovala tudi v mobilnih napravah v realnem času. Metoda se je sposobna izogniti tudi rekonstrukciji nepravilnostim prekrivnega popačenja iz nizkoločljivostnih slik. Uporablja sistem, ki temelji na učenju. Sistem se je učil na več tisoč parih nizko- in visokoločljivostnih slik, tako da zna rekonstruirati nizkoločljivostne podrobnosti [30].

Program bi lahko razširili tako, da bi dodali tudi možnost superresolucije s frekvenčno-domensko metodo in regularizacijsko metodo. To bi nam odprlo tudi več možnosti za še boljše rezultate.

Literatura

- [1] S. Anthony. CSI-style super resolution image enlargement. (julij 2012). *Extreme tech* [Online]. Dostopno na: <http://www.extremetech.com/extreme/132950-csi-style-super-resolution-image-enlargment-yeeaaaah>.
- [2] D. Glasner, S. Bagon in M. Irani. Computer Vision Lab Super-Resolution From a Single Image. (2009). *Faculty of Mathematics and Computer Science* [Online]. Dostopno na: <http://www.wisdom.weizmann.ac.il/~vision/SingleImageSR.html>.
- [3] M. Borengasser. Introduction to Forensic Image Processing. (junij2014). *Forensic Mag* [Online]. Dostopno na: <http://www.forensicmag.com/articles/2014/06/introduction-forensic-image-processing>.
- [4] Image processing. *Wikipedia* [Online]. Dostopno na: https://en.wikipedia.org/wiki/Image_processing.
- [5] D. Alvar. Can you get that license plate? (oktober 2014). *Amped blog* [Online]. Dostopno na: <http://blog.ampedsoftware.com/2014/10/09/can-you-get-that-license-plate/>.
- [6] G. Anbarjafari. Digital image processing. *University of Tartu* [Online]. Dostopno na: <https://sisu.ut.ee/imageprocessing/book/5>.
- [7] vReveal. *vReveal* [Online]. Dostopno na: <http://www.vreveal.com/>.
- [8] Ikena Forensic Video Enhancement software. *Motion DSP* [Online]. Dostopno na: <https://www.motiondsp.com/ikena-forensic/>.
- [9] Amped FIVE features. *Amped* [Online]. Dostopno na: <https://ampedsoftware.com/features>.
- [10] GIMP – Feature Overview. *GIMP* [Online]. Dostopno na: <https://www.gimp.org/features/>.
- [11] GIMP – Wikipedia. *Wikipedia* [Online]. Dostopno na: <https://en.wikipedia.org/wiki/GIMP>.
- [12] C. Baron. Adobe Photoshop Forensics: Sleuths, Truths and Fauxtography. *Cengage Learning*, 2008 [Online]. Dostopno na: <https://books.google.si/books?id=1c8LAAAAQBAJ>.

- [13] M. G. Roque, R. M. Musmanno, A. Montenegro in E. W. G. Clua. Adapting the Sobel Edge Detector and Canny Edge Extractor. (2010). *Instituto de Computacao* [Online]. Dostopno na: http://www2.ic.uff.br/iwssip2010/Proceedings/nav/papers/paper_161.pdf.
- [14] S. Price. The Canny Edge Detector. (julij 1994). *The University of Edinburgh – School of Informatics* [Online]. Dostopno na: http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/MARBLE/low/edges/canny.htm.
- [15] Tadej Žgur. Detekcija položaja kroglice igralniški ruleti s pomočjo računalniškega vida. *Univerza v Ljubljani – Fakulteta za računalništvo in informatiko*, Ljubljana, april 2005, str. 18-19.
- [16] Edge detection. *The University of Auckland* [Online]. Dostopno na: https://www.cs.auckland.ac.nz/courses/compsci373s1c/PatricesLectures/Edge%20detection-Sobel_2up.pdf.
- [17] Other Methods of Edge Detection. *Rice Web Services* [Online]. Dostopno na: <http://www.owlnet.rice.edu/~elec539/Projects97/morphjrks/moredge.html>.
- [18] Noise removal. *Mathworks* [Online]. Dostopno na: <https://www.mathworks.com/help/images/noise-removal.html>.
- [19] E. Weisstein. Gaussian function. *Wolfram Mathworld* [Online]. Dostopno na: <http://mathworld.wolfram.com/GaussianFunction.html>.
- [20] Gaussian blur algorithm. (november 2012). *Pixelstech.net* [Online]. Dostopno na: <http://www.pixelstech.net/article/1353768112-Gaussian-Blur-Algorithm>.
- [21] J. Ludwig. Image Convolution. *Satellite Digital Image Analysis - Portland State University* [Online]. Dostopno na: http://web.pdx.edu/~jduh/courses/Archive/geog481w07/Students/Ludwig_ImageConvolution.pdf.
- [22] R. Fisher, S. Perkins, A. Walker in E. Wolfart. Spatial filters – Unsharp mask. (2003). *The University of Edinburgh – School of Informatics* [Online]. Dostopno na: <http://homepages.inf.ed.ac.uk/rbf/HIPR2/unsharp.htm>.

- [23] Unsharp mask. *Diffraction Limited* [Online]. Dostopno na: https://diffractionlimited.com/help/maximdl/Unsharp_Mask_Basics.htm.
- [24] F. G. Loch. Image Processing Algorithms Part 4: Brightness Adjustment. (2008). *DF Studios* [Online]. Dostopno na: <http://www.dfstudios.co.uk/articles/programming/image-programming-algorithms/image-processing-algorithms-part-4-brightness-adjustment/>.
- [25] F. G. Loch. Image Processing Algorithms Part 5 : Contrast adjustment. (2008). *DF Studios*. [Online]. <http://www.dfstudios.co.uk/articles/programming/image-programming-algorithms/image-processing-algorithms-part-5-contrast-adjustment/>.
- [26] J. C. Ferreira. Algorithms for super-resolution for images based on sparse representation and manifolds. *Image Processing. Universite Rennes I*. November 2016, str. 29-34, Dostopno na: <https://tel.archives-ouvertes.fr/tel-01388977/document>.
- [27] Matlab. *Mathworks* [Online]. Dostopno na: <https://www.mathworks.com/products/matlab.html>.
- [28] M. Irani in S. Peleg. Super-resolution from image sequences. *Department of Computer Science – The Hebrew University of Jerusalem* [Online]. Dostopno na: <http://www.cs.huji.ac.il/%7Epeleg/papers/icpr90-SuperResolutionSequences.pdf>.
- [29] D. Lancaster. A review of some pixel interpolation algorithms. (2007). *Tinaja* [Online]. Dostopno na <http://www.tinaja.com/glib/pixintpl.pdf>.
- [30] M. Zhang. Google's RAISR uses machine learning to enhance low-res photos. (november 2016). *PetaPixel* [Online]. Dostopno na: <https://petapixel.com/2016/11/16/googles-raizr-uses-machine-learning-enhance-low-res-photos/>.